Applications of Neural Networks to Traffic Monitoring Equipment Accuracy and Predictability

By:
C. James Elliott, Group EES-5
Jason Pepin, Group X-NH
of Los Alamos National Laboratory

and

Ralph Gillmann
Office of Highway Information Management
Federal Highway Administration

In cooperation with the

Alliance for Transportation Research
1001 University Blvd. SE
Suite 103
Albuquerque NM 87106-4342

Project funded by:

Federal Highway Administration
Office of Highway Information Management
400 Seventh Street, S. W.
Washington, D. C. 20590

Submitted by:

Earth and Environmental Sciences Division
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official view or policy of the Department of Transportation or the Federal Highway Administration or Los Alamos National Laboratory. The mentioning of a product or manufacturer in no way constitutes an endorsement thereunto.

January 1997

Applications of Neural Networks to Traffic Monitoring Equipment Accuracy and Predictability

C. James Elliott(EES), Jason Pepin(XDIV), Ralph Gillmann (FHWA)

Earth and Env. Sciences Division/ Applied Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545

Final Report

We have utilized ground truth and vendor data from the 1993 Georgia I20 data set as the basis of two studies into the accuracy of vendor traffic monitoring equipment and a description of the methods. We performed detailed examination of characteristics of erroneous classifications obtained by vendors after improving the ground truth data set and aligning vendor events with the ground truth video-based events. The results of this study were displayed on an hour-by-hour basis and revealed that equipment undergoes intermittent or Byzantine behavior as to its ability to correctly classify. The majority of these errors occurred in Class 9, the eighteen wheelers. A by-product of this study was two means of instantaneously diagnosing vendor equipment performance by examining the axle spacing data for abnormal behavior. Our second study was to determine the ability of historical data to assist in prediction of classification. This study assumes Byzantine behavior is corrected. We showed that not only representative data could be utilized to make substantial improvements by factors of 1 to 10 in classification accuracy, but also data taken on May 5 could improve prediction on all six days of the study by factors of up to 10 for some two-axle vehicle classes and factors of 3 for some non-two-axle classes. Part of the methodology involved constructing a neural network classifier for all 13 classes of the FHWA scheme. With this, we made improvements to standards now in place for vehicle classification. We also give a detailed description of a simple neural network and include three papers describing technical details that include the classifier, a new tractable neural network and measures of capacity.

classification, classifier, data editing, dichotomy, fault, fault analysis, perceptron, probabilistic neural network, sparse neural network, tractable neural network, traffic monitoring, traffic monitoring equipment, Vapnik-Chervonenkis distance, vehicle neural network

Executive Review

Prologue

This work was performed by a Work for Other Federal Agencies agreement proposal that included a scope of work. The scope of work set out seven tasks, one of which is producing this report, all directed toward the goal of improving the traffic monitoring process. (1) The first task involved a detailed examination of the Georgia Tech Research Institute (GTRI) Interstate-20 Georgia data set. (2) The second task was to construct a neural network to classify all 13 of the FHWA classes. (3) The third was to port the artificial neural network to a PC computer and compare performance to Scheme F using the ground truth portion of the data set as judge. (4) The fourth was to distill the logic into an "if-then" algorithm and compare the performance to Scheme F. (5) The fifth was to develop validation criteria using the neural network approach. (6) The sixth was to use the validation method on vendor data. (7) The seventh was to participate in efforts to develop vehicle classification and other travel monitoring standards. (8) The eight was to produce this report addressing those tasks. This report contains an executive summary including conclusions and recommendations, documentation and summary of all tasks, an evaluation of the performance of the techniques used, and a description of the neural network in layman's language. In addition to the official requirements, seven additional tasks were accomplished: (I) Expand the description of neural networks to give a basic understanding in layman's language of training and testing of data using a neural network. (II) Develop a technique to analyze faults in the Georgia I20 data set in hourly bins including a graphical representation of the reasoning. (III) Produce a paper and give a talk to a neural network conference about time synchronization effects in video ground truth at two different collection sites. (IV) Produce a formal document describing Byzantine (intermittent) effects in traffic monitoring equipment based upon the GTRI study. (V) Produce a formal document showing that a fundamental limitation of many neural networks can be overcome by designing a new algorithm. (VI) Document the entire process in a hierarchy of levels, the highest of which is the agreement compliance, the next the executive summary, individual summaries of work in three areas: the I20 study, the neural network descriptions, and work related to standards; the next level is to include relevant source articles extracted from monthly progress reports and/or formal descriptions of the work. The latter includes among other items an introduction that places the work in the context of other work in the field and an abstract of the work in formal terms; this work, however, is in some cases at a more technical level. (VII) Port the neural network software to an AIX-compatible computer.

The task compliance is addressed by topic. (1) The completed ground truth data set was sent to the sponsor in FoxPro and ascii formats. Details of the processing are given the in-depth directed review of the I20 project. (2) The completed neural network was described in a paper entitled Beyond Scheme F delivered at the NATDAC conference; many other details are given in this report. (3) The port of the artificial neural network was to an IBM PC computer; the performance of the neural network *vis-a-vis* Scheme F was also reported in the paper Beyond Scheme F. (4) The logic distillation (if/then) was accomplished by several approaches; the one judged most appropriate was the modification standard E1572 of the American Society for Testing and Materials (ASTM), improving its performance on the GTRI data where the neural network had performed better. Performance changes using this modified standard along with an expectation edit process are described as calibration methods. (5)The validation criteria for classification were developed at two levels: (a) a Go/No-Go approach for the axle spacings and (b) utilization of a modification of the neural network reported in Beyond Scheme F. These results are documented among other places in a formal paper entitled Detection of Catastrophic-Fault-Prone Classification Data. (6) The application to vendor data is also described there. (7) The section of the report entitled Potential Classification Schemes addresses advanced concepts for modification of standards such as are considered by committee E-17 of the ASTM. (8) This paragraph documents the performance of those enumerated tasks and this report documents the results thereof.

Report Structure

The structure of this report is multilevel. The first level is the overall or executive summary. It includes the objectives of the study and the three parts of the study each with conclusions/recommendations. The next level is a detailed directed review including references to articles and papers to follow. It includes references to articles from the monthly publication Traffic Monitoring Progress and three papers, one of which was published in the proceedings of the National Traffic Data Acquisition Conference (NATDAC) ‘96 conference. This level is intended to describe what details are given in those reports/papers expanding upon the description given in the previous level. The excerpted articles and papers to which reference is made then follow as the last major level. The papers themselves, however, have abstracts which serve as an additional level for that genre. Depending on his purpose, the reader is expected to pick and choose the portions of this report. Even the in-depth directed reviews are intended to be readable without interruption as an overview of what further descriptions are available. A second reading of the in-depth directed review can then serve as the basis of the directed study coupled with hopping back and forth to the references. The technical level of the executive summary is lowest, increasing with depth of level. Some terms, e.g. Byzantine, become increasingly refined with depth where, for instance, possible causes of Byzantine faults are discussed in the incorporated paper entitled *Detection of Catastrophic-Fault-Prone Classification Data*. In the case of articles that belong to more than one of the parts of the study, the name of the other study will be mentioned. For instance, the *Nearest Neighbor Classification Applied to Georgia Data for 4 Axles* article naturally falls in the In-Depth Review of Neural Networks part but is mentioned in the *In-Depth Review of Georgia I20 Study* part, but not in the *In-Depth Review of Potential Classification Schemes* part. The papers are at a level appropriate for publication within their field; thus, the abstract and introduction of a paper are intended for a wider technical audience than the body of the paper. The material in the *In-Depth Review of Georgia I20 Study* is a directed study starting with the refining of the data set, proceeding to show how the classification scheme was improved, and culminating with a description of the Byzantine faults that limited performance of the equipment studied. The material in the *In-Depth Review of Potential Classification Schemes* part is intended for those interested in new classification schemes. It includes material on improving the axle classification (AC) method known as American Society for Materials and Testing (ASTM) E1572 and material on the calibration techniques we developed also called hybrid schemes. The material in the *In Depth Review of Neural Networks* part describes the details of the neural network aspects of the work. It is intended to answer not only the question of what work was done but also that of how it was done. The level of presentation is aimed at a non-scientific college graduate level. The paper entitled *A Tractable Sparse Probability Neural Network* remains at this level for its abstract and introduction but requires an understanding of computational algorithms in its body. Because of the length of the papers they are arranged at the end of the directed study sections, the articles from the progress reports are placed after them in chronological order by the month in which they appeared.

Finally, several references may be noted. The Traffic Monitoring Guide (June 1985, U. S. Department of Transportation, Federal Highway Administration, Office of Highway Planning, Washington, DC, 20590) contains a description of the 13 Classes used by the FHWA for classification. Scheme F is described in the paper *Beyond Scheme F* in Table 3 and references; it is not a scheme approved by the FHWA.

Study of Georgia I20 Data

The Georgia Tech Research Institute (GTRI) staff, led by Bruce Harvey, collected and analyzed data taken in Georgia on Interstate 20, within the outskirts of Atlanta in May and September of 1993. The GTRI staff followed the guidelines of specific tasks. They produced data of two varieties: vendor data and extracts from video data that we call the ground truth. We found and removed a number of outliers and other errors from the data set obtained on 3.5 in. floppy disks. This modified data set was delivered to the FHWA in ASCII and FoxPro formats. The corrections were fewer than 1000 and, although their direct incorporation into the study would not have altered the thrust of the GTRI conclusions, in making these corrections, we were concerned that the errors might be causing effects involving a small number of vehicles. Examining the errors also led us to study the accompanying video and to discover many unusual vehicle types (also appearing in this report). One of the major problems in analyzing the data set was the establishing of a correspondence between the vendor clocks and the clock used in the video. The video camera was located opposite the direction of traffic flow from the vendors by 260m (850ft) at most. The GTRI staff found an intrinsic error between these clocks of approximately 3 seconds which could not be removed by any ordinary means. We found that by using a technique called super synchronization, we could lower the size of the error substantially. This technique involved using the frame number of the video to subdivide the second into 30 parts. We do not emphasize this method because electronic time stamps can provide better information directly on the video. However, the paper titled “Super Synchronization for fused Video and Time-Series Neural Network Training” may be a help in understanding the interaction of the geometric and temporal aspects of video frames. The paper, which was not included as part of the final report, was presented at the Neural Network Applications in Highway and Vehicle Engineering conference, April 10 and 11, 1996, The George Washington University, Ashburn, Virginia.

The recommendations and conclusions we reached from the I-20 study are about the experiment and its results. They include the following:
(1a) Research conducting any future study involving video cameras should consider recording the time on the video electronically to fractions of a second. (1b) The vendor time should be obtained to an accuracy of 10 to 100 millisecond. This higher precision for both the vendors and the ground truth would improve the process of data collation by time.
(2a) A computer-assisted method of analysis of the video could further improve the data set reliability, reduce the processing cost, and possibly remove the necessity for the electronic time

stamp.
(2b) Video contrast control and camera feedback adjustment can improve the ground truth data, making it more amenable to computer processing.
(2c) Qualitative operational definitions of classes such as Class 5 (having four tires on the second axle) that are determined by visual clues should be documented; or a quantitative procedure such as determining the number of tires on an axle using a slanted piezo detector as is done in Australia should be adopted.
(2d) In any traffic-monitoring experiment, lane-changing effects should be planned for and either incorporated or removed from the data set.
(3) Substantial improvements in the ability to classify vehicles are masked by Byzantine (intermittent) errors.
(4) Byzantine errors can be detected as described in the paper “Detection of Catastrophic-Fault-Prone Classification Data.”
(5) If the conclusions generalized for the event error result in the I-20 study, a Class 9-based fault scheme can be constructed and incorporated as a vendor algorithm in order to detect a variety of these Byzantine equipment failures.
(6) Dropped axles do not occur at random; we have established patterns of dropped axles for Class 9 vehicles particular to several vendors. Such patterns can be used by manufacturers to establish norms for their equipment in the category of standard faults reported. Deviations from these standard faults can still occur by the Byzantine mechanism.
(7) Any axle-spacing-only classification scheme with better performance than E1572 should be incorporated as a modification to that standard, or a standard with a similar determination of vehicle structure should be adopted.

Potential Classifier Schemes

Although an original intent of this study was to improve potential classifier schemes (PCS), Byzantine faults became an overriding problem. Assuming that the Byzantine fault problem can be ameliorated and that the GTRI study is representative, we believe that this original goal can also be achieved. If the measurements are error-free, the accuracy of the information contained in the performance matrix for vehicle identification can be enhanced by varying amounts depending on the class: up to 10 times for at least one class and up to 3 times for several other classes. These results were proven by a very stringent test that incorporates one day of the May data to predict the September results. A substantial part of this improvement occurs for two-axle vehicle classes and, as indicated, relies on historical data. The if/then analysis task led to suggested recommendations for improvements to a well-known standard in addition to the improvements based on historical data. Improving this standard was possible, as a result of the better performance of the neural network classification scheme (for some classes). The key question in that analysis was the following: Exactly what constituted simplicity? It became evident that expectations about vehicle structure could be incorporated beneficially in a simple description. These expectations are about the appropriateness of particular axle-spacing parameters and whether there should be differences in determined parameters under various assumptions such as regarding the end of all trailers as the same. These approaches have to do with the structure of the vehicle. The question was asked: How can we incorporate the vehicle structure into the classification method? Our answer builds upon the classification scheme known as ASTM E1572 or the axle classification code. This scheme has built into it a determination of the vehicle structure. The ultimate if/then scheme, thus, builds upon a scheme such as E1572 or something very much like it. We allowed E1572 to determine the gross part of the structure (including the number of trailers or semitrailers), thereby reducing the classification to a set of much smaller problems. It is not yet clear whether the changes obtained to the standards for the Georgia data will apply elsewhere.

In conclusion, substantial potential exists for improvement to current classification schemes by two means: (a) use of historical data calibration techniques and (b) modifications to a standard such as ASTM E1572. The three elements of a system, albeit expensive, that relies on historical data are: (1) an accurate, automated, portable means of determining vehicle ground truth to provide accurate historical data; (2) a site-based historical data accumulation procedure using the portable equipment;and (3) use of a calibration technique that processes the historical data. Wherever the Georgia data apply, we can improve classification by 3 to 10 times on select classes with this system and standard operating procedures. Ideally, these classification algorithms would self-diagnose to indicate when the need for recalibration arises. However, pavement management systems require axle-loading information and would not benefit substantially unless such data were also included. We recommend that any axle-spacing-only classification scheme with better performance than that of E1572 be incorporated as a modification to that standard or to a standard with a similar determination of vehicle structure. Those interested in studying the issues further or in proposing an improved classification method will be interested in the *In-Depth Review of PCS.*

Neural Networks

The neural network development effort proceeded through several phases. We started this work with an emphasis on the nearest-neighbor type of algorithm. Such an algorithm was an approach that captured the essence of the problem. However, it suffers from the drawback of requiring n^2 operations, where n is the number of training points. The nearest-neighbor method also has approximate symmetry built into it, a concept discussed further in the directed *In-Depth Review of Neural Networks*. For an on-line vendor classification operation or for fast computer calculations, a faster algorithm is needed. The seminal idea for this faster algorithm was described at a conference, organized by Dr. Morton Oskard of FHWA (Turner-Fairbanks complex) in a presentation by Dr. Azim Eskadarian. First, the algorithm is faster because it required only a fixed number of operations per training sample. Second, we replaced the technique known as Cerebellar Model Arithmetic Computer, better known as CMAC, with a fuzzy interpolation technique. This process allowed the algorithm's property known as generalization to be quite similar to the nearest-neighbor technique. The algorithm also produces the probabilities of each of the domains into which the parameter space is divided; this type of system is known as a probability neural network (PNN).

We tried other classification methods on the data set, including decision trees with a variety of pruning techniques. An unusual variety of decision tree, known as an oblique plane tree, was also utilized. The object was to arrive at a simple if/then analysis of the data. The details of this process are described in the *In-Depth Review of Neural Networks*.

The simple description of the neural network that we provide deals with two fundamental questions: (1) What is the power of a neural network? (2) How can one train a neural network? We do not answer these questions in general ourselves. Rather, we show the power that a specialized neural network has in being able to compose an arbitrary fax such as is used to send images by telephone. The idea of how to train the neural network parameters to produce that fax was not answered by this idea. We simply showed that a set of parameters exists which, if the neural network could be so trained, would produce the desired fax. The topic of training is taken up for a set of points, each of which is in one of two known classes. When these points can be divided into two classes by a dividing plane, a single-layer perceptron can train its weights to give the correct classification of all points. In our discussion of perceptron training, we show how and why the perceptron learns by example without using any algebra.

Then two papers follow that we have produced: (1) “A Tractable Sparse Probability Neural Network” (SPNN) and (2) “Detection of Catastrophic-Fault-Prone Classification Data.” The first paper is an outgrowth of the PNN discussed earlier. This paper is concerned with the problem of arriving at a neural network that can compute its weights in a reasonable amount of time when there are many input variables used to train the neural network. With such a capability it is desirable to avoid having the computation time increase geometrically with the number of input variables, as is known to occur for perceptrons and back-propagation networks. This SPNN network removes the geometrical-dependence bottleneck occurring in the first neural network that we designed with fuzzy interpolation. It will, therefore, be appropriate when axle-loading inputs are specified in addition to axle-spacing inputs, thereby more than doubling the number of input variables.

The second paper is concerned with a means to diagnose traffic monitoring equipment undergoing Byzantine (i.e., intermittent) faults. It shows that for a particular data set, the faults can be so catastrophic that the condition can be diagnosed without the corroborating data called the ground truth. The paper also shows that the conventional measure of neural-network capacity, called the Vapnik-Chervonenkis distance, is an inappropriate and misleading measure; the label distance or number of nodes used in the algorithm is a more appropriate measure.

In conclusion, we have developed powerful techniques using neural network methods that are suited to the analysis of problems even more complicated than the ones we have already addressed. We have placed these in the context of the neural network literature to which these techniques have contributed.

Table Of Contents

PAPERS

ARTICLES FROM TRAFFIC MONITORING PROGRESS

List Of Figures

PAPERS

Beyond Scheme F

Detection of Catastrophic-Fault-Prone Classification Data

ARTICLES FROM TRAFFIC MONITORING PROGRESS

Extracted Georgia-I20-Study Articles

Extracted Potential-Classification-Schemes Articles

Extracted Neural-Network Articles

List Of Tables

PAPERS

Beyond Scheme F

A Tractable Sparse Probability Neural Network

ARTICLES FROM TRAFFIC MONITORING PROGRESS

Extracted Georgia-I20-Study Articles

Extracted Potential-Classification-Schemes Articles

Extracted Neural-Network Articles

In-Depth Review of the Georgia I20 Study

Preliminary analysis of the Georgia Tech Research Institute (GTRI) database of Interstate I20 vehicles from May and September 1993 revealed a number of phenomena. In the June 1995 Traffic Monitoring Progress report, an article entitled *Georgia Data Set Has Outlier Points*, file names, statistics, and some GTRI methods were presented. The July report entitled *Georgia Data Has Vehicles with Strange Axle Separations -- The Outliers* shows some of the video images corresponding to outlier data points. Details about specific outliers are described in categories refined with respect to the number of axles and the vehicle class. In the August article “FoxPro Database Edited and Sent to FHWA”, we described technical points about differences in terminology and previously unexplained techniques used in classification. We also point out that operational definitions are useful for classes that are not uniquely defined by the axle-spacing data. For instance, the height of a vehicle being larger than a predetermined amount, a flared-out rear wheel skirt, or an indented hub were operational definitions for a Class 5 vehicle, which requires the 4 tires on its second axle that are unobservable in the video. In the *Vendor Data Arrives* article the top ten reasons the number of events in the vendor measurements file do not correspond to those in the video ground truth file are explained. The nearest neighbor classification scheme applied to the 4 axle data is described in the *Neural Network* part. This scheme can also serve as the metric for judging other schemes that can be computed more quickly. In the article entitled *Vendor Data Phenomena,* appearing September 1995, reasons related to lane changing that produce a lack of correspondence between vehicles in the vendor and ground truth data sets are discussed. Time errors and speed errors are also discussed. The article entitled *Split-Vehicle Lost Axle Count* describes another fault and suggests that understanding it must come from the vendor; we can only speculate as to what is happening without the vendor insight to the internal operation of his classifier. The question of lack of temporal correspondence between the ground truth and vendor values is explored in the article *Downstream Effects* and four possible reasons are given. The *Super Synchronization* articles of September and October further explore these temporal effects. The assumption of slow time drifts of all the clocks permits analysis leading to intrinsic uncertainties as low as 0.1 seconds. This occurs at the expense of counting video frames, carefully modeling downstream motion, and observing discontinuities in time-delay graphs. The video camera is situated near the first vendor station but is 260 m away from the last vendor. Inconsistencies that show up in the vendor sites located far from the camera are associated with uncertainties in the length of the vehicle. These can translate into uncertainties in the time of up to one second. Does the vendor measure the time of an event when the front of the vehicle crosses his first sensor or the back of the vehicle crosses the last sensor? This topic also is described in the proceedings of the previously mentioned neural network conference [Neural Network Applications in Highway and Vehicle Engineering, April, 1996]. The September article entitled *Jerry Schiff’s Observations* gives the vendor point of view of the Georgia study.

The Performance Matrix Evaluated for Scheme F appearing in October 1995 describes results of a classification scheme commonly used and modified by vendors of vehicle classification equipment; Scheme F is not a standard approved by the FHWA. We show how Scheme F performs relative to the ground truth, had it been adopted by the vendors as their classification scheme. A major problem with Scheme F appears in the guise of large asymmetries in the off-diagonal matrix elements. The nearest neighbor classification schemes have more balanced off-diagonal matrix elements and thus Scheme F appears remiss in this regard. The balancing of off-diagonal matrix elements removes errors of prediction by making, on average, the Class 2 vehicles classified as Class 3 exactly cancel the Class 3 vehicles classified as Class 2. Further details are described below in the directed review of the paper entitled *Beyond Scheme F* and the December article entitled *Symmetry of Nearest Neighbor Performance Matrix*. In the latter article, idealized assumptions are laid out that attempt to be reasonable approximations. They produce sufficient conditions to achieve symmetric matrix elements. Another set of perhaps less likely assumptions that lead to asymmetric elements for the nearest neighbor classifier are also given.

We describe how a typical vehicle is classified in the December 1995 article entitled *Reinventing the Parser*. The parser referred to takes time-stamped sensor events and decides how to separate the event stream into individual vehicles. This process is analogous to taking word events of sentences and turning them into meaningful information. The vehicle event parser can involve a time block stream of P1, P2 or L1, L2 events such as P1P1L1P1P2P2P2L2P1P1P1P1. Some of these are noise events like “ahs” and “ers” of an uncertainly formed sentence, but they are detected by observing when their amplitude (loudness) is weak. The piezo events, P1 and P2, are assigned a time corresponding to the threshold of when an event rises above the noise level. That event is the pressure impulse (later transformed into an electrical impulse) impinging on the linearly configured piezo sensor stretched across and embedded in the highway. For the loop events L1 and L2, the time of first signal sensed is utilized as if stamped on a time card. This time corresponds to the front of the vehicle moving over the leading edge of what is often a 2-m by 2-m loop. If you are interested in the details of rules used to analyze the events, you are urged to read the article for an exhaustive description of the analysis of 35 time-block streams. An accompanying article entitled *Parser Tool Aids Analysis* is background material for why a parser tool is needed. Originally, we had intended to continue the description of this tool into the subsequent month, but it was omitted. The idea of the tool was to plot vehicle events in a two-dimensional space with the abscissa or horizontal axis representing time and the vertical axis representing distance down the road. An event was represented graphically by a colored circle whose width and color indicated the amplitude of the signal. Related but spatially separated events are tied together by colored lines. These related events are of two varieties: (1) the activation of two piezo detectors, one downstream a known amount from the other, by the same tire. (2) the events of the front (back) of the vehicle crossing the leading edge of the first loop and the front (back) of the vehicle crossing the leading edge of the next loop. In the case of the loop, what constitutes these events is ambiguous as alternative interpretations are discussed in the articles. For a passenger car in an alternating loop piezo system with two loops and two piezos, the lines have a slope related to the vehicle speed, and all four lines should appear parallel to each other. Missing sensor events are not paired with corresponding events, in which case the number of lines may be three or less. Noise events are not paired either, and these appear as small circles with colors corresponding to small amplitude.

At the Neural Network Applications in Highway and Vehicle Engineering conference held at the Ashburn Virginia campus of The George Washington University in April 1996, Azim Eskandarian described the utilization of a hash table in a neural network called the Cerebellar Model Arithmetic Computer. This idea led to our Sparse Probability Neural Network (SPNN) presented at the NATDAC conference in the paper *Beyond Scheme F* (reviewed below) held in Albuquerque in May 1996. The concept of a hash table and the ideas of SPNN are described in the Neural Networks part. In this part we concentrate on the aspects relevant to the study of the Georgia data as presented in the paper.

The paper *Beyond Scheme F* included in this report begins by pointing out that Scheme F can be represented by a McCulloch-Pitts neural network. This type of neural network is credited as progenitor of the entire field of neural networks. The details of its operation is described there for three axle vehicle classification. The definition of Scheme F is also given there. Using Scheme F and the edited Georgia I20 data we computed a performance matrix. This matrix is simply a table of information telling how each of the vehicles in a given class become classified with a particular classification scheme. The results, of course, depend on the historical data used. The form of the results for vehicles in a given class are the number of vehicles classified into each of the classes. In the May 5-7 table, all but 5 of the 3676 Class 9 vehicles were classified as Class 9. The 5 misclassified vehicles were labeled as Class 11. Data were also presented for the September 9-11 period. Characteristic of Scheme F is the lack of cancellation of errors between Classes 2 and 3. Of 8565 Class 2 vehicles, 243 were called Class 3, and 8316 were called Class 2. Of the roughly 5500 Class 3 vehicles, 3018 were called Class 2 and 2443 were called Class 3. The asymmetry between the 243 vehicles and the 3018 vehicles is glaring. If this asymmetry did not exist, errors arising from the misclassification of Class 2 as Class 3 would cancel on average those of Class 3 being labeled Class 2. Scheme F apparently was created to produce the most likely vehicle identification for a given set of data. If for a particular set of axle spacings using a particular historical data set, 51% of the vehicles are Class 3 and 49% are Class 2, the most likely approach calls the vehicle Class 3. A fuzzy logic approach, or probability approach, says that the vehicle has at 51% membership in Class 3 and a 49% membership in Class 2. Although a given vehicle is in one class or the other, in a statistical sense it is much more accurate to add up vehicle memberships as if fractions of a vehicle were tallied. This method of aggregation is incorporated in SPNN which we have suggested is expected to be much more accurate than Scheme F.

The membership numbers for SPPN classification of the Georgia data are generated by choosing 1/3 of the data at random as the historical training set. The two other sets of 1/3 of the data each are used to generate the performance matrix. In one of the tests, Class 2 has 4703 correctly identified as Class 2 and 1548 incorrectly identified as Class 3. The Class 3 numbers have 2214 correctly classified for Class 3 and 1579 incorrectly classified as Class 2. On a statistical basis, the 1548 vehicles and the 1579 vehicles almost cancel in tallying either Class 2 or Class 3, thereby improving the count accuracy in each class for periods of observation involving many vehicles. A detail about the SPNN method is that it will not classify a vehicle unless it has seen a similar one in the historical data. To handle the unclassified vehicles, we developed another SPNN that was much more generous in deciding what vehicle was its neighbor. All the rejects of the first SPNN are reclassified by the second SPNN. This arrangement is called cascading. The details of the performance for other classes are described in the paper *Beyond Scheme F.* Since that work another method called calibration was developed in July that further improves performance for all classes. In the May 1996 article entitled *Georgia Data Analyzed*, we describe the replacement of the FoxPro-based algorithm with a C++ algorithm. The FoxPro calculations required 4-6 hours to run as contrasted with 5 seconds required for calculations described in the *Beyond Scheme F* paper. With the new program, we were able to make rapid improvements to the algorithm for splitting vehicles that had been two vehicles masquerading as one. The algorithm that had taken the 4-6 hours per run in FoxPro on a PC now took a turn-around time of 50 seconds on a Sun Sparc 10. In these runs the effect of distance downstream from the ground truth camera are easily seen by the number of exceptions produced by the program. At the vendor site 90-m downstream there were 7 exceptions whereas at 200-m downstream there were 500 exceptions, and at 260-m downstream there were 800 exceptions. The exceptions were primarily lane-changing events that failed to appear in the ground-truth database. These were more likely to occur the farther downstream the measurements were made. We discussed the rationale for believing why these events were lane changing or passing events and not vendors reporting extra vehicles. Incidental examination of these data sets resulted in the first reported observation of Byzantine (intermittent operation) faults.

In July 1996, we utilized an new method of predicting performance based on historical data that we described in the article entitled *Performance of the Calibrated Georgia Data Set*. Here calibration means training the neural net with historical data by a particular technique; this technique is described in the In-Depth Review of Neural Networks part below. Within this technique are several methods for training the network. The training data may be thought of as the statistical sample upon which the predictions are predicated. Method I takes a representative sample from the entire database. The sample is selected at random. Method II uses the events from a particular day (the first day unless stated otherwise). We expect the Method I results to be a good prediction of vehicle counts for all the days. How well Method II works for predicting events on days other than the sample day depends on unknowns related to the traffic dynamics and habits of drivers in the part of the world of the survey. The ability to extrapolate predictions is called generalization ability. Thus, Method I has no test of generalization capability because the predictions are of a representative;, whereas, the day-based Method II is being tested for its ability to generalize to an unknown time period.

We compare the calibration methods to Scheme F, the baseline method which is taken to represent typical performance of vendor equipment. Unlike Scheme F, the calibration method gives almost perfect predictions when tested on its own training data; only Class 12 has an error above 1% and that is 4%. This method is also referred to as the hybrid method for reasons described in the In-Depth Review of Neural Networks part. We use 1/3 of the data to train and 1/3 to test; the calibration method has at most 15% to 20% errors (excluding Class 7 because of too little data). This contrasts with Scheme F which has well over 100% errors. Tables are given class by class comparing the hybrid method to Scheme F on a day-by-day basis having trained on random data. On average, of course the day-by-day results follow the previous results of 15%-20% errors. But especially classes with a small number of measured vehicles have large day-to-day fluctuations. Class 4, predicted by the hybrid method for instance has the percent errors of 9,-20,56,-11, and -37 in contrast to the Scheme F errors of 116,112,200,96,260,137, and 36. When training with the May 5 data, tabulations are also given. The hybrid method was a factor of 3-4 better on Classes 8 and 11. It was almost an order of magnitude better on prediction of Classes 2 and 3 that performed at the 97% and 94% level respectively. Additional calibration rules could improve the results even further. Data are also presented using September 9 data to predict September 10 and 11. Two pictures are presented of rarely seen vehicles.

Byzantine (intermittent operation) errors were hinted at in the report entitled *Accuracy of Traffic Monitoring Equipment* produced by co-workers at GTRI. Some acknowledged Byzantine faults were present in equipment malfunctions such as latching up of processor boards. The report states “Of the eleven classifiers installed, five were malfunctioning; three were repaired on-site and two were returned to the vendor facility for repair.” No task had been set forth for GTRI to examine the time dependence of the errors.

Our *Fault Sensing Overview* article of August 1996 is of interest for someone wanting to obtain a theoretical idea of where faults might be expected. Faults are characterized as being persistent or intermittent. The time scales involved did not directly anticipate the Byzantine scale of hours we later observed. The phenomena of dropped axles was described in August and continued in the September article both entitled *Dropped Axles for Class 9*; there interesting patterns of missing axles are given by vendor.

In these early August studies of fault sensing, the concept of “otherwise rare events” is used. This term has been dropped because the events are not rare; they do, however, masquerade as rare events but only when considered individually. Collectively, there are too many of them. The term we now use is *faults* or *catastrophic faults*, the latter being specifically those faults that lie outside the normal range of perfect data. Those faults that lie within the normal range are called *contained faults,* an example being a compact car erroneously having the axle spacing of a large pickup. The dropped axle theme is picked up in the September 1996 issue

The August 1996 article entitled *Vehicle Reject via PNN* gives the plan to study faults. The plotting of the data versus time binned by hour was described there. Only two such graphs per vendor showing total events and rejected events, however, were planned. The article entitled *A Class 9 Test?* points out a perplexing condition of having 10%-88% errors of all Class 9 classifications, whereas the classification scheme was better than 99% accurate. The classification methodology was, thus, eliminated as a possible explanation for those errors. What was left was vendor faults.

Two accounts are given of the vendor faults known as Byzantine errors characterized as such in the later account. The first account has more data and is the September 1996 article entitled *Separation Fault Sensing*. Both accounts describe the two methods used. The first method was to maintain the essence of the idea in its simplest form: a series of Go/No-Go tests. The second method was to obtain the most precise measurement possible of the phenomena of catastrophic faults using a probabilistic neural network (PNN). The description we give here is of the second account which is a subset of the first account. Those interested in the details of the results of the three vendors are urged to read the full account below. The second account is the paper entitled *Detection of Catastrophic-Fault-Prone Classification Data.* We do not describe all the material in the paper’s introduction that places the problem in context of other work. The introduction does characterize faults according to the work of Andrzej Pelc in network theory. In network theory at least five terms are used: permanent or transient, crash fault or Byzantine fault, and fault prone. The first two terms follow their ordinary usage, where permanent indicates the entire time period under observation. The term transient also includes what we call intermittent. The term crash fault is for a system that behaves either perfectly or crashes and thereby provides no information as if the on/off switch was turned off. A Byzantine fault describes arbitrary (even malicious behavior), it can include stopping, rerouting, or altered messages. Fault prone is a state of equipment that produces many faults. The range of normal data is treated differently in the two cases. For the Go/No-Go method, each axle spacing has a range of allowable, non-fault values. These are bounded above and below. If the N axle spacings of a vehicle fall outside the range of any of the corresponding Go/No-Go ranges, it is considered a catastrophic fault. If all the possible points (potential vehicles) in the box-like region in parameter space described above corresponded to real vehicles there would be little need to improve upon it. However the actual range of the second to third axle spacing depends on whether the vehicle has 4 or 5 axles, for instance. Effects of this variety can be diagnosed and treated by carving space up into little regions such as squares or cubes, but they cannot be described by the method outlined above called Go/No-Go that has but one such region. In a space of higher dimension than three, the cubes still exist mathematically but are difficult to visualize. All of space is carved up into such cubes. Mathematically, each cube has a smaller Go/No-Go range of its own for each of the N spacings, and any set of N axle spacings fall into one and only one cube. These cubes are taken to be adjacent to each other so that any set of N axle spacings appropriate for the first method falls into one of the cubes of the second method.

Then, ignoring the small detail of generalization described in the paper, both methods have a Go/No-Go test for each of the N axle spacings. The only difference being that in the first method there is only one cube. In the second method (not called Go/No-Go) there are lots of smaller cubes each with their own such tests. Any such cube imposes a test on the vendor data only if occupied by a ground truth data point. When a cube is occupied, it corresponds to a ground truth vehicle with axle spacings that gives a Go condition for that cube. If the vendor data falls in a cube that has not been activated to the Go state, it is considered a fault. Five graphs plotted in hourly counts are presented. They show (a) hourly count of all vehicles, (b) the neural net hourly rejects (they do not fall into a Go cube), (c) the separations that are out of bounds as determined by the simple Go/No-Go method, (d) the faults as determined by the ground truth, and (e) that fraction of the faults that have ground truth identity of Class 9. More data is available in the July 1996 section of Extracted Neural Network Articles.

The observations are that (1) the faults are a small fraction of the total count (2) faults occur at hour 30, among other times, during a low count period for total vehicles, (3) the rejects for all classes are very similar in shape to the Class 9 faults and are of greater amplitude, (4) the simple separations out of bounds counts are less than half the neural net rejects, (5) the ground truth differences are larger in amplitude than the neural net rejects, (6) the half-width of the peak faults is about 10 hours.

These observations give rise to the corresponding conclusions: (1) the faults can be hard to detect, (2) the faults can be a significant fraction of all counts, (3) most of the catastrophic faults are Class 9 faults, (4) the simple Go/No-Go method is about half as sensitive as the neural net method, (5) contained faults occurring for two axle vehicles are perhaps 25% of all faults, (6) the Byzantine catastrophic faults come and go, being more likely at night.

In the October 1996 articles entitled *Power Quality Issues* and *Line Power Monitoring*, various standards that relate to emissions immunity are given. These point the way to experimental methods of detecting Byzantine behavior.

In conclusion, a major goal of obtaining accurate counts requires removal of the Byzantine counts of both the catastrophic and contained varieties. This being done, there is the potential for more accurate count predictions by using calibration methods that depend on historical data.

In-Depth Review of Potential Classification Schemes

We describe two ways we can improve current classification schemes. The emphasis is on the ideas behind the two methods. The first is an improvement to the American Society for Testing and Materials (ASTM) standard E1572. The second uses the hybrid or calibration scheme the results of which were reported in the In-Depth Review of the Georgia I20 Study part. The eternal challenge is to incorporate any major classification improvements into a new standard or to modify an old standard. Ultimately such classification standards rest on a historical database that includes records of individual vehicles. Each vehicle record in the ground truth set includes two parts (1) a set of measurements that serve as the basis for the classification prediction and (2) the true identification class of the vehicle. The ground truth set used in this study is the Georgia I20 Data of 1993. The study is ideal in the sense that many vendor effects (giving errors of special character) are not included because here we use only the ground truth. This is best because the error type can vary from vendor to vendor. However, the relevance of this approach is predicated on either the experimental removal of Byzantine errors from vendor equipment measurements or the utilization of ground truth data.

The improved performance achieved by the sparse probability neural network (SPNN) is leveraged by using it to critique the E1572 standard and pointing out where that standard can be improved. Then by analysis of the differences between the higher performance network and E1572, changes in the parameters of E1572 are made. To be able to achieve the analysis, we must understand how E1572 works. This functioning is described by the usage of decision tree notation (see In-Depth Review of Neural Networks part) in the July article entitled *AC Code Modifier Code*. We explain the concepts used in the algorithm including what the structure code means in the article *E1572 FORTRAN Algorithm*. The details of the improvement procedure are described in an article entitled *Improving the AC Code?* We now give an example of the kind of improvement that can arise by this type of methodology. With E1572, Class 5 is wrongly identified as Class 4 a total of 53 times versus approximately 12 for SPNN. The AC code has two separate levels which could cause the problem: (1) the structure determination and (2) the modifier determination. The structure determination gives the information about how many units there are and how the axles are grouped together on those units. It is the physical representation of the vehicle. The modifier describes how to go from a given structure to the FHWA class designation as one of the 13 classes; it also depends on a controlling parameter. In the case of the Class 5 vs. Class 4 distinction, both have the same structure: a single unit with two axles. The difference has to be in the modifier. In this case the controlling parameter is called TWOMAX with an E1572 value of 6.1-m (20’). Increasing the value to 6.7-m (22’) drops the 53 errors to 3 at the expense of a single new error: a Class 6 vehicle erroneously labeled a Class 4. Other examples are given in the article of corrections involving Classes 3 and 8 and another involving Classes 5 and 8. Before proposing these as changes to the standard, however, they need to be checked at other locations throughout the country for good performance; i.e., we need to check the generalization. Even if they do not generalize, an adaptation procedure for these parameters would allow them to fit a given the classification at a given locale.

The calibration or hybrid scheme can be looked at as a new way to store information for vehicle identification. In the raw form of the method, two-axle vehicles are classified by a means different from the other vehicles. An augmented E1572 standard is used for vehicles with more than 2 axles. The two-axle data is binned into 100 10-cm bins. The summary form transforms the raw information into a prediction based on historical data. This method has been further improved and is called a fuzzy method in the article entitled *Improved Fuzzy Method*, but we include it in the class of calibration or hybrid schemes now viewing the fuzzy characterization as a detail. Here, the first 5 classes which describe vehicles with 2 axles in most cases, are split up into the ten classes: those with exactly two axles and those with more than two axles. This bifurcation of the classes removes the highly volatile vehicles with more than two axles (e.g. towing trailers) from the otherwise more stable population of 2 axle vehicles. These new classes can be virtual in the sense that the final classification can still use the 13 Class scheme. They are simply chosen to be a better set of predictor variables that are tracked in accumulating historical data. The results of this calibration method were reported in the In-Depth Review of the Georgia I20 Study part.

In-Depth Review of Neural Networks

In this review we discuss the topics of nearest neighbor networks, McCulloch-Pitts networks, perceptrons, decision trees, hash tables, probability neural networks, and sparsity. References are provided here to articles extracted from the publication entitled Traffic Monitoring Progress appearing as part of this final report in the section entitled *Extracted Neural Network Articles*. Nearest neighbor classifiers are discussed in the article entitled *Nearest Neighbor Classification Applied to Georgia Data for 4 Axles* appearing December 1995. Technically, this type of classifier can be considered a feed-forward neural network. What it does can be explained easily for two dimensions using the concept of graph paper. The two variables with information upon which the classification prediction is based serve as the abscissa or x-axis and ordinate or y-axis coordinates. Each item to be classified is represented by a point on the piece of graph paper. The point is colored red for the red class and blue for the blue class. The training data then appear as a number of red points and a number of blue points on the graph paper. Now suppose one has test data with unknown color that needs to be classified red or blue. One simply plots the test point on the graph paper (but does not color it) and finds the colored point on the graph paper that is closest to the test point and uses its color; that is what is meant by nearest neighbor. In higher dimensions it is the same principle. For 4-axle data, there are three axle-spacing numbers, and the data are points in three dimensional space with different colors for each FHWA class.

The McCulloch-Pitts classifier was the original neural network. It was originally suited to what was called nervous activity. We describe a special case of such a classifier consisting of one computational element conventionally called a neuron. The neuron we describe has two inputs, two weights, one output, and a threshold. Its behavior is simply to give an output signal of 1 if the weighted sum of its inputs exceeds the threshold; if not, it gives an output of -1. This idealized unit when hooked together with other such units and appropriate weights has the power to describe any binary function. An example of such is any (black and white) Fax. Such neurons can also have a range of outputs more like a grey scale printer than a Fax. For more details on how a Fax is computed see the article *How A Neural Network Works Geometrically* appearing in the Spring 1996 section.

The perceptron classifier received a lot of attention as a single layer device. Its inputs were combined in the same way as the simple McCullogh-Pitts classifier. Single layer means that many neurons can be used but they all attach to the same inputs but have only one output line each. Such a layer can be analyzed neuron by neuron as they do not interact. A single neuron perceptron also has a training rule that enables it to adapt to the input in a fashion that is characterized as learning. It is this ability to train that captured the imaginations of many investigators and added one more step to the doorway of the burgeoning edifice of neural networks. What a perceptron can do is train to classify any set of red and blue points on graph paper that are linearly separable. All red points must fall on one side of a line and all blue points on the other. Then the training rule starts with any line anywhere and adapts it to serve as a classifier giving 100% correct results. The same idea works in three or more dimensions, and a proof of this well-known theorem is given, without using algebra, in the October 1996 article entitled *Training a Single-Layer Perceptron Classifier.* The proof is written at the level of a non-technical college graduate with substantial spatial visualization skills and a flare for abstract thinking.

In early 1996 we became acquainted with the concept of a hash table. Hash tables are very common in computer operating systems. They are typically used when one types a command name to be executed. Rather than have a table of all possible combinations of letters, the name is hashed or transformed into a number. The number is limited in range, perhaps 1 to 1000, and many potential names map to the same number. However, the mapping is pseudo (artificially) random so that any given name obtains a corresponding number that appears to be a random choice of the numbers between 1 and 1000. If two names happen to have the same number, unlikely as that may be, a collision is said to occur. Special handling called a linked list is used to referee collisions. Once a number is obtained, it can then invoke the command such as Print. The concept of a hash table addresses the question of efficient storage of names and/or numbers, an issue of importance in dealing with large partially filled arrays of data called sparse arrays.

The concept of a sparse probability neural network (SPNN) can be described by ignoring all detailed questions of storage of information. That storage is handled by the hash table. In two dimensions, each square of the graph paper corresponds to a range of x values and a range of y values. Typically, the x value would be the first axle spacing and the y value the second spacing. The choice of which ranges of these spacings to use is up to the user of the network. Typically the right hand side of the graph paper will represent the maximum of the x values and the left hand side will represent the minimum. Likewise for the y values, the minimum and maximum correspond to the top and bottom of the page. Suppose we are trying to classify three-axle vehicles. Then for the two spacings, we can take the abscissa to be the first spacing of the axles and the ordinate to be the last spacing. The spacing corresponding to the distance between the third axle and the first can be determined from the other two. It need not be considered because it adds no new information. For training the neural network, assuming three classes of such vehicles: red, yellow, and blue, we place colored dots in the corresponding squares so that we can count them. The corresponding square is determined from the pair of numbers representing the axle spacings of a vehicle in the training set. At the end of processing all training vehicles, if most squares have no vehicle dots in them the table is called sparse. The table of probabilities is made by counting the dots of each color in the square, finding the total number of dots in the square, and converting each count into an estimate of the probability of finding a dot in that square, i.e., the number of dots of one color divided by the total number of dots. The probabilities are written in each occupied square, one in each color there. On testing, the test vehicle has two axle-spacings that fall into a particular square on the graph paper. If no probability numbers have been recorded there, the vehicle is said to be a reject. If probability numbers are there, the test vehicle is said to have membership for each of the colors represented. For instance, if equal numbers of red, yellow, and blue dots were in the square in question, the test vehicle would be given 1/3 membership in each of the classes. Those membership fractions are estimates of the probabilities of the vehicle belonging to each of the classes. The SPNN answers the question: what are the membership fractions for a given test vehicle? A traffic counter using an SPNN would tally up all such fractions in determining the vehicle counts for each class. These concepts are described in more detail in the paper entitled *Beyond Scheme F.* There a second SPNN is also constructed to handle the reject vehicles. The reject classifier uses graph paper with many fewer squares for the same ranges. The full SPNN described there also handles vehicles with any number of axles. It also distributes training points to the four nearest squares rather than to be in one square exclusively. For instance, if a point fell on the corner of a square, it would be equally distributed to the touching squares with membership 1/4 each.

The paper entitled *Tractable Sparse Probability Neural Network* adds a new twist. The simplest kind of tractable SPNN has already been described. It is the case where each point goes into only one and only one square. The lack of tractability has to do with the amount of computation time required to distribute the membership of a training particle such as the one falling in a corner. In two dimensions tractability is not an issue. But with many dimensions it is an issue. Just like the doubling of the number of grains on the successive squares of a chess board, for each new axle spacing added, the number of touching regions doubles. Not only axle spacings can serve as new dimensions, but also the weight loading of each axle can be a new variable. Other quantities such as the height of the vehicle or its length could also serve as new variables. For problems with a large number of variables, the computational time becomes an unbearable load. The remedy described in the tractable SPNN is to abandon splitting the membership in a fashion that doubles or grows geometrically with each new dimension and replace it with one that grows arithmetically with each new dimension. The computation time is in analogy with placing the grains. On a chess board, with the new rule of adding two grains to each square, there would be only 128 grains rather than 9,223,372,036,854,775,808 grains by doubling on each square having started with 1 grain.

The paper entitled Detection of Catastrophic-Fault-Prone Classification Data uses an SPNN but colors all the dots black using the ground truth training data. Then it uses the vendor data and sees if it is rejected as a fault. Exactly how this works on real data has already been described in the In-Depth Review of the Georgia I20 Study part. Another aspect of that paper is relevant to this neural network part. That aspect has to do with the question of how to measure the capacity of a neural network that uses only black and white colors.

The capacity of a neural network with black and white output is conventionally defined as the number of training points with black and white colors that can be guaranteed to correctly classify themselves by the method. This number is called the Vapnik-Chervonenkis (VC) dimension. The problem is that this definition is not appropriate to the SPNN network used for fault prediction. In two dimensions in particular, we are trying to fit a Fax, discussed earlier, having square pixels, whereas in the standard approach the function class is unlimited. First let us look at the number of points that can be guaranteed in our neural network, after we train the network, to correctly classify fault (black) or no-fault (white) vehicles upon testing. The standard approach requires the consideration of all possible locations for all the points and all possible colors (black and white) for the points. Thus in the standard approach, all the points could fall within the same box although they had different colors. In that case, because of the rule that each box is trained to be either black or white, the points cannot all be correctly classified. However, by shifting the graph paper back and forth, or by making the squares larger, the maximum number of points that can be guaranteed to correctly classify themselves is 2 or 3. Special cases of the points having the same value of x need not be considered because they rarely happen. But all points that can fall within the same box cannot be dismissed as unimportant by the rules because they involve a measurable area of points. We know that we cannot train our SPNN network to function well with only 2 or 3 points. We could handle regions like a solid circle or a solid triangle with our neural network and lots of other regions, but such training would require hundreds of points. Why is the VC capacity only 2 or 3? We propose a way around this dilemma by proposing another measure of capacity that gives the capacity as the number of boxes on the Fax (looked at as if graph paper). This measure is obtained by adding a new rule. The new rule states that if any training point falls in a box, all other training points that would fall within the same box with the opposite class are eliminated from the training set. The rational for this rule is that the box size is chosen to represent a region where it is known ahead of time that all the points that fall within it will have the same class. Why should the capacity measure be penalized by the requirement that we consider points of opposite class in the same box even though we know that they will not occur?

Decision trees is a part of what we call if/then analysis. The term if/then is taken to indicate a simple explanation of what is happening. The larger the number of logical decisions required in the logical description, the less well we consider it to meet the term of if/then analysis. A decision tree in its simplest form has two branches forming from the trunk. A single decision is used to decide which branch is to be used. Each branch can be divided into two sub-branches by another single decision rule. This process can extend to an arbitrary number of sub-branches. The branches or sub-branches without further sub-branches are said to be leaves of the binary tree. Thus the fewer the branches, the better the simple explanation. The if/then analysis is then to find the simplest tree. Another factor that constitutes a simple tree is the complexity of the decision at each branching. A simple decision rule contributes to the overall simplicity of the tree. Decision trees are sometimes called a sieve. A neural network is a decision tree of a particular variety of decision rule.

The usual class of decision trees has a single variable compared to a threshold at each decision point. These are called axis-parallel decision trees. We can think of an electrical analogy where the input variable is being fed by a wire to the decision fork. Thus a decision tree has a number of wires hanging down to the ground, one wire for each decision. Different decisions can use the same input variables; thus, more than one wire can be gathered together at the ground plane for that input feed. Such cases are said to have *fan out* of the input variable. Unlike a multi-layered neural network, where the input wires go only to the first layer, decision trees have input wires going to all levels. Each fork in the decision tree can then be modeled as a neuron in a neural network with one input coming from an input feed and another input coming from a previous level. We are beginning to see that an axis parallel binary decision tree can be represented as a neural network with neurons having only two inputs.

The training of an axis parallel binary decision tree involves choosing the input variable and the threshold level at each node. This is usually done by a deterministic algorithm that is computed once. In the training process, the training data may be thought of as being fed into the trunk of the tree. At the first decision node, some of the training data goes one way, and the rest of it goes the other way. At each sub-branch, a subset of the data resides. Based on this limited amount of training data, a variable is chosen and a threshold level is set that divides the training data in two. This is done by using some measure of how well the classes are split at the node. Another kind of a decision tree is an oblique decision tree as described in the two June 1996 articles entitled *If/Then Analysis of the Georgia Data Set* and *Are Oblique Planes Understandable?* These kinds require every input to be attached at each decision node. Furthermore each of the inputs has its own weight at each node. The complication is that the weights have to be trained in addition to threshold selection. One way to train the weights is to use a perceptron. A perceptron uses only two classes. Various classes can be pooled together until there are only two classes which are trained.

The perceptron method is not usually used here probably because of the lack of uniqueness in pooling the classes. In principle all possible combinations of pooling the classes into two superclasses could be used with a criterion for choosing the best combination. Hopefully, some mathematical simplification would occur in such a procedure that would make the algorithm tractable. Even if it were possible to obtain a tractable oblique decision tree, it probably would not be used much because it is not a simple axis-parallel tree and does not lend itself to obtaining an unobliterated understanding what is going on.

In the article of July 1996 entitled *Decision Tree Classification Methodology*, the need for understanding the underlying mechanism in the decisions and having that reflected in the tree is pointed out. These decisions for trucks are made by the manufacturers of the trucks. Several examples are given.

Finally, we discuss a topic from the driver-decision point of view that is of relevance to the underlying mechanism in traffic flow; it is discussed in the July 1996 article entitled *Parameter Estimation*. Typically these decisions are looked at from an origin-and-destination point of view. Many of these decisions have elements for many drivers in common such as taking children to school or going to a place of work. Some of these common decisions are predictable, others remain hidden such as plans to go to a private party or to cut wood from a particular forest. Other hidden plans could involve holiday deviations from normal patterns or attendance at a popular movie. The question arises of how to model such a variety of behavior for a single site. An example of a model that may be viewed as a combination of fuzzy logic and a single-neuron neural network with training is called a probability model. It has two classes of vehicles with triangular membership functions of the axle spacing. The second triangle keeps its shape but is allowed to translate its center to larger or shorter axle spacing values, thereby modeling hidden changes in plans that could be occurring. Four methods of finding the parameters of this model are ranked best to worse (1) bin by bin edit matrix, (2) weighted least squares, (3) demarcation line edit matrix near error balancing point and (4) demarcation line at error balancing point. See the July article for details. To study this kind of model experimentally, binned axle data needs to be accumulated. Origin-and-destination simulations could also give insights.

Beyond Scheme F

C. James Elliott, Henry Fisher, and Jason Pepin

Los Alamos National Laboratory

and

Ralph Gillmann, FHWA

Abstract

Traffic classification techniques were evaluated using data from a 1993 investigation of the traffic flow patterns on I-20 in Georgia. First we improved the data by sifting through the data base, checking against the original video for questionable events and removing and/or repairing questionable events. We used this data base to critique the performance quantitatively of a classification method known as Scheme F. As a context for improving the approach, we show in this paper that scheme F can be represented as a McCullogh-Pitts neural network, or as an equivalent decomposition of the plane. We found that Scheme F, among other things, severely misrepresents the number of vehicles in Class 3 by labeling them as Class 2. After discussing the basic classification problem in terms of what is measured, and what is the desired prediction goal, we set forth desirable characteristics of the classification scheme and describe a recurrent neural network system that partitions the high dimensional space up into bins for each axle separation. The collection of bin numbers, one for each of the axle separations, specifies a region in the axle space called a hyper-bin. All the vehicles counted that have the same set of bin numbers are in the same hyper-bin. The probability of the occurrence of a particular class in that hyper-bin is the relative frequency with which that class occurs in that set of bin numbers. This type of algorithm produces classification results that are much more balanced and uniform with respect to Classes 2 and 3 and Class 10. In particular, the cancellation of errors of classification that occurs is for many applications the ideal classification scenario. The neural network results are presented in the form of a primary classification network and a reclassification network, the performance matrices for which are presented. Cancellation of errors is site dependent and the question of generalization to many sites suggests another measurement of goodness for a classification scheme that takes into account the need to operate the network under a variety of operating conditions.

Introduction

An unofficial standard for the classification of axle-spacing-based highway data into the thirteen classes of the Federal Highway Administration was developed by the Maine Highway Department a decade ago and is known as Scheme F. This scheme is not a FHWA standard and presents a highly inaccurate count of Class 2 and Class 3 vehicles in the Georgia study. We show that the assumptions built into this method can be overcome by adaptive means or by using probabilistic neural networks. Furthermore, these more general methods of characterizing partitioned axle parameter space, yield higher precision results.

Regions of parameter space requiring special attention are those regions where two or more of the 13 FHWA classes can not be uniquely separated. The Scheme F approach rigidly split parameter space into fixed compartments that were declared by fiat to be one and only one of the 13 classes. It was recognized that this approach was not strictly correct, but one would hope that the boundaries could be carefully modified so that the errors incurred in one region of parameter space would cancel those in the other regions. For this to be the case, a number of assumptions have to hold. One of these assumptions is that the distribution of passenger cars (Class 2) and pickup-up trucks and vans (Class 3) remains of invariant shape from place to place. Another is that the relative numbers of Class 2 and Class 3 vehicles conforms to a fixed ratio. Common sense opposes these assumptions as does extensive ground truth data. We utilize the 1993 study of traffic on I20 near Covington Georgia to demonstrate that the assumption of cancelling errors under scheme F does not hold, even remotely.

The I20 data suggest new approaches to the axle-based vehicle classification process. One of these is application of a probabilistic neural network that assumes static distributions. This network is tested with two data sets both of which have good cancellation properties. We also discuss adaptive classification procedures, the simplest of which is the moving demarcation line model. This model moves the linear boundary between two classes depending on observations. This method is highly non-linear in that it involves ascertaining the characteristics of the flow from all of the vehicles before deciding how to partition up the ambiguous regions of parameter space. Although the subclass decomposition is often well behaved, we point out instances where the concept of axle spectra has overwhelming experimental difficulties preventing its realization. Practical data-reduction difficulties that occurred in acquiring the Georgia data are also discussed.

Utilizing the Georgia data, we describe a new neural net approach, reclassification of the rejected vehicles by a specially designed network for that purpose. Combined with the adaptive approach described above for the overlap regions, this approach promises greatly enhanced generalizability that tailors itself to particular geographic sites and/or functional categories.

Scheme F results

Scheme F is a method developed by the Maine Department of Transportation [Wyman, 1984] for the classification of vehicles. The basic regions for Scheme F are shown in Table 1 for each of the 13 Classes of the FHWA classification. Scheme F has not been approved by the FHWA as a method of classification nor has it been based upon a systematic analysis of an underlying database that supports it as an optimal data classification scheme. We hold that any modern classification scheme must meet the criterion of conforming well to an established database. The results of classification using Scheme F of the Georgia I20 database are presented in Tables 2 and 3. These results closely track the relative performance matrices reported by Harvey for manufactured equipment in his study. That is, it appears that many vendors are basing their classification scheme on something that closely resembles Scheme F. The lack of symmetry of errors can be seen by examination of the vehicles misclassified as Class 2 while they are really Class3. These are compared to the Class 2 vehicles that have been classified as Class 3. The latter are in the hundreds whereas the former are in the thousands. This type of asymmetry is characteristic of Scheme F shown in Tables 1 and 2. Another problem is the misclassification of Class 10 vehicles as Class 13. The definition of Scheme F that we used is shown in Table 3.

Next we point out that Scheme F can be formulated as a McCulloch-Pitts neural network [McCulloch, 1943]; an example of this formulation is shown in Figure 1. Here we look at all the 3 axle vehicles. They can fall into Classes 2,3,4, 6 or 8. The McCulloch-Pitts neural network has two input nodes; one for the separation between axles 1 and 2 and the other for axle separations 2 and 3. We give an example and trace how the neural net responds to inputs of 12 and 14 respectively.

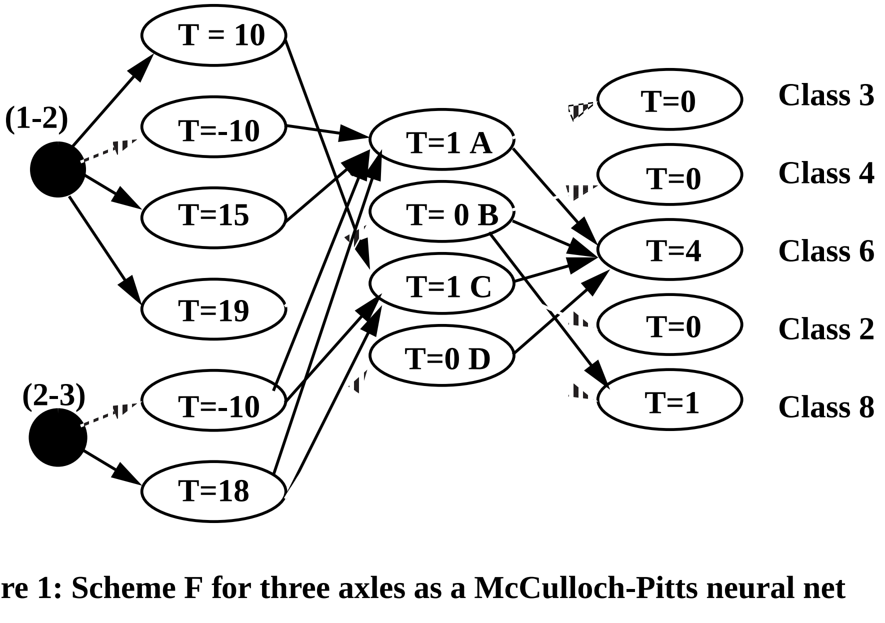


Figure 1: Scheme F for three axles as a McCulloch-Pitts neural net

With 12 for the first axle separation (1-2), only those nodes that are above threshold fire. The dashed line indicates a negative weight of -1. Thus 12 becomes -12 going into the t=-10 node. Because -12 is below -10, the second node in the first layer does not fire. The first node in the first layer has a threshold of 10 which is below the value of 12, and thus it fires in the above threshold

Table 1: M_{ij} for Scheme F for the May 5-7, 1993 Georgia test

Truth\Scheme F	1	2	3	4	5	6	7	8	9	10	11	12	13
1	18	2				1							
2		8316	243			1		5					
3		3018	2443		4	3		87			5		
4			3	60	6			1					
5		22	361	93	235	1	9	22			16		
6			3	20		461		1	14	1		1	
7													
8			3					319					
9									3671		5		
10										31			12
11								3	1		101		
12												29	
13													1

Table 2: M_{ij} for Scheme F for the September 9-11, 1993 Georgia test

Truth\Scheme F	1	2	3	4	5	6	7	8	9	10	11	12	13
1	30	3											
2		10128	234			2		9					
3		3700	2495		44	3	1	79			3		
4			2	44	5	4							
5		7	145	62	149		6	18		1	7		
6			2	10	1	191		3	5				
7							1	2					
8		2	6			1	1	278					
9									3152	1	11	1	
10										27			9
11											93		
12												28	
13													

Table 3: The Scheme F assumptions

				Axle	Space	In	Feet
Class	FHWA Vehicle Type	No of Axles	Axle 1 to 2	Axle 2 to 3	Axle 3 to 4	Axle 4 to 5	Axle 5 to 6
1	motorcycle	2	0-5.8				
2	car	2	5.8-10				
	car/1 axle trailer	3	0-10	10-18			
	car/2 axle trailer	4	0-10		<3.5		
3	pickup	2	10-15				
	pickup/1 axle trailer	3	10-15	10-18			
	pickup/2 axle trailer	4	10-15		<3.5		
	pickup/3 axle trailer	5	9.9-15			<3.5	
4	bus	2	>20				
	bus	3	>19				
5	single unit truck/dual rear axle	2	15-20				
6	single unit truck	3		<18			
7	single unit truck	4					
8	2 axle tractor with 1 axle trailer	3		>18			
	3 axle tractor with 1 axle trailer	4		<=5	>10		
	2 axle tractor with 2 axle trailer	4		>5	>3.5		
9	3 axle tractor with 2 axle trailer	5					
	2 axle tractor with 3 axle trailer	5		<6.1		3.5-8	
10	any single tractor/trailer comb. with 6 or more axles	6 or more			3.5-5		
11	any tractor/double trailer unit with 5 or less axles	5		>6			
12	tractor/double trailer unit	6					>10
13	any tractor/double trailer unit with 7 or more axles	7 or more					

condition. It is the only node that fires in the first layer that is connected to the first axle separation value. The second axle separation has 14 as the input. For the first of two nodes to which it is connected, the weight is -1 as indicated by the dashed arrow; thus the value going into the t=-10 node is -14, a value that is below threshold. Thus the first of the two nodes in layer one attached to (2-3) does not fire. Neither does the second node with t=18 because 14 is below threshold. For these values, the only node that fires in the first layer is the first node. When a node fires it produces a value of unity; otherwise it produces zero.

We now examine our special case for the second layer neurons that are denoted by A,B,C, and D. Of these, B,C, and D fire but A does not. The neuron C fires because it is connected to the firing neuron of layer 1. Neurons B and D fire because they have thresholds of 0 and have 0 inputs. Neuron A does not fire because none of its inputs are active. The interpretation of these nodes may be helpful in understanding the remaining part of the net: Node A fires whenever Class 3 is not active. Node B fires whenever the second axle separation is less than 19. Node C fires whenever Class 2 is not active. Node D fires whenever the second separation is less than 18.

The third layer is the classification layer. The reader may establish a few facts: Class 2 occurs whenever node C does not fire. Class 3 occurs whenever A does not fire. Class 4 occurs whenever B does not fire. The Class 6 and Class 8 nodes rely on the fact that the neural inputs sum the signals entering the node. Each of the lettered nodes emit either a zero or unity pulse; only if all four inputs leading to Class 6 fire will the threshold of 4 be achieved. Because of the negative weight of -1 for the D node leading to Class 8, its edge can contribute only zero or -1 to the sum, whereas the edge from the B node can contribute either zero or unity depending on its firing condition. For this case only, if B fires and D does not, will the Class 8 node occur.

Although the language of how a neural network operates is more difficult for most humans to follow in detail, its description, as given above, is nearly in the form required for a computer to absorb. Humans have a highly developed visual cortex and pictures are much more to their liking than the set of 15 rules computers like so well. In Figure 2, we show the same three axle system as a picture with the shaded areas representing regions where the Scheme F classification occurs. The axle separation (1-2) is plotted as the ordinate and the separation (2-3) is plotted as the abscissa. The scale can be established by a comparison with Table 1. Figure 2 contains the same information as Figure 1, but it is in a much more digestible form.

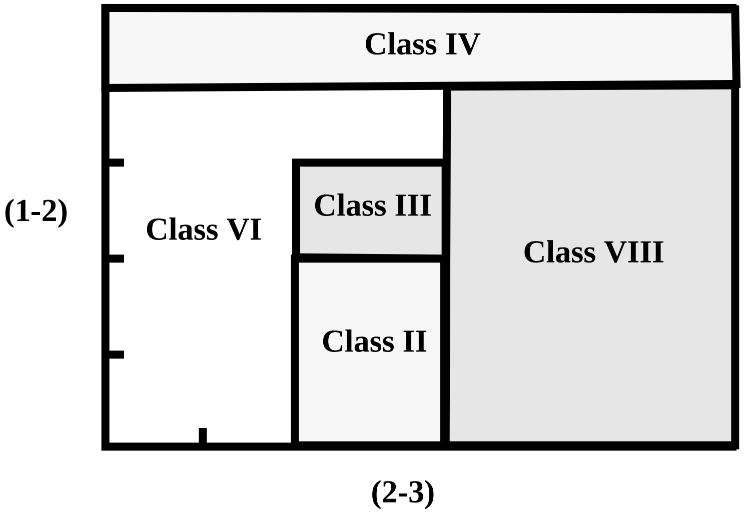


Figure 2: Scheme F for three axle vehicles visually

We can see that a neural network partitions a plane up into black and white regions that correspond to whether a particular node fires or does not fire. Although this is not quite in the form required for a classification scheme, it is very close. The scheme we describe later involves dividing each axle spacing up into bins, thereby dividing the plane up into a set of rectangular (hyper-bin) regions. In Scheme F, if properly defined, the hyper-bins fall into one - and only one - of the 13 classes. In the general scheme described later, known as a probabilistic neural network, each of the hyper-bins has a mixture of the classes. For a given data set, a particular hyper-bin could contain, for instance, 10 vehicles in Class 2 and 34 vehicles in Class 3.

The Neural Network Approach

The basic information that we measure on the highway can be cast in terms of the hyper-bin in which the axle separations of the vehicle falls. If two vehicles fall into the same hyper-bin of bins with bin numbers (4,3,5,0,0,0) their axle separations are considered to be equal within the context of our model. We use the convention of bin 0 for the i'th axle separation of a vehicle when that vehicle has too few axles to generate that separation. Thus, the example vehicle described is a four axle vehicle with three axle separations. All vehicles having these bins for its axles will be compared against each other. If a hyper-bin of bins has only one class of vehicle, it can be assigned to that class uniquely. The most general situation is for a hyper-bin of bins to have more than one type of vehicle. Accumulation of the number of each type of vehicle that falls into a given hyper-bin of bins occurs during the training phase of the neural network. These numbers are then converted into a probability (more correctly a likelihood) of finding the vehicle in that hyper-bin of bins. In the testing phase, a vehicle that falls into a given hyper-bin of bins is equally considered to be any of the vehicles that have fallen into that hyper-bin during the training phase. Thus, a probability for each class of the 13 classes of the FHWA is assigned to the test vehicle according to these numbers. In the course of determining the performance matrix during the testing phase,

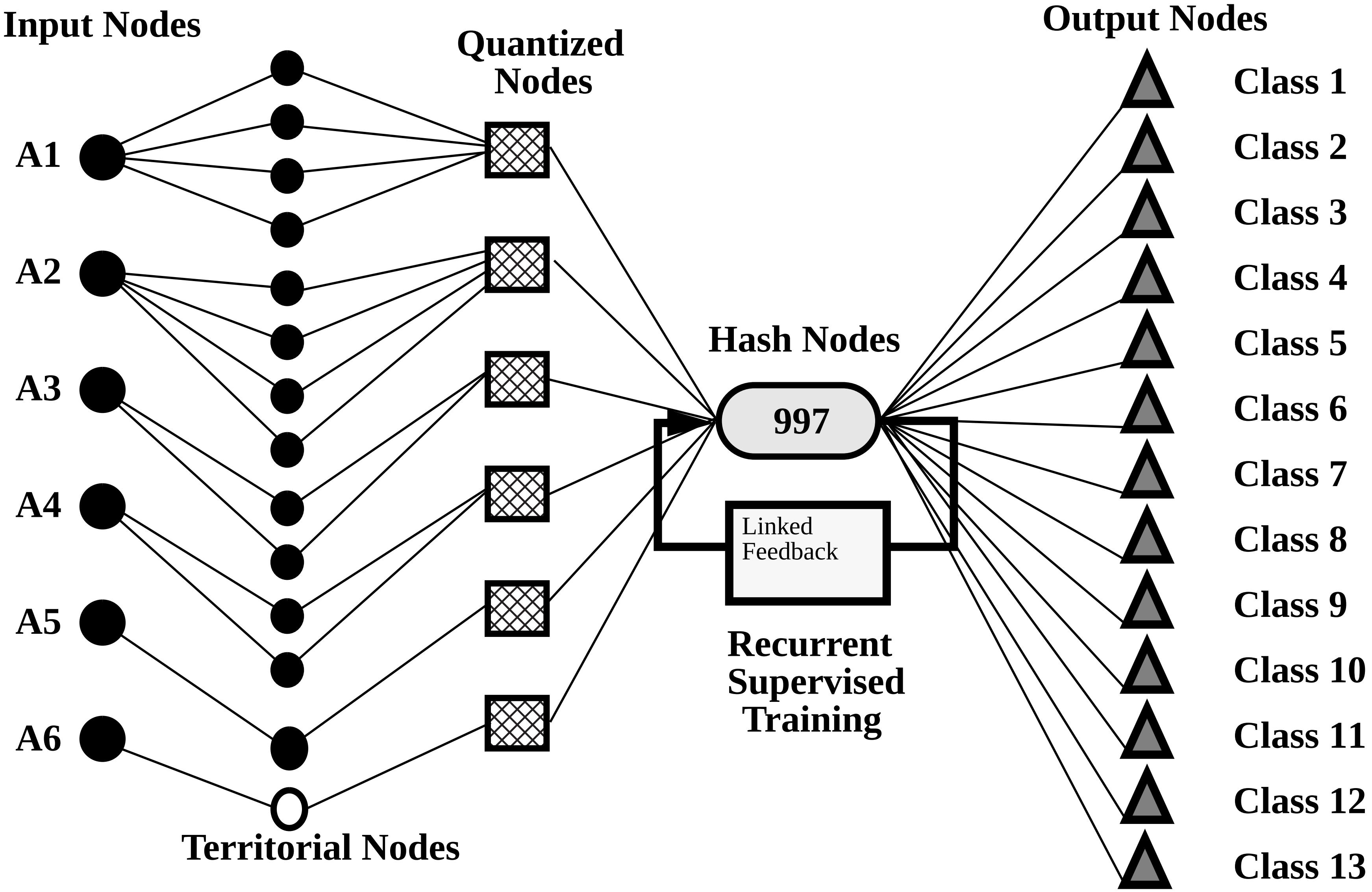


Figure 3: The Probabilistic Neural Network Architecture

the performance matrix element is incremented according to the probability.

The neural network has six axle separations as the inputs. Each axle separation has a range of continuous values whose lower range is greater than zero. The value of zero is used to indicate that the vehicle is missing such a separation. The continuous range of separations is broken up into a user determined number of bins. A technical problem in achieving such a system is the potentially large number of bins that can occur. For the example below, the number of bin combinations is 3.2 million. The computer science technique of a hash table [Sedgewick, 94] is used to condense the number of possibilities to only those hyper-bins that actually have vehicles. This is approximately a thousand such hyper-bins, a savings of about a factor of 3000 in the number of storage elements that could be accessed as the storage bins. The net classification time is less than the input and output time - less than 6 seconds total on a SUN SPARC 10.

Table 4: Axle Spacings in 30.48 cm units

axle spacing	min space	max space	number of bins
1	4.0	30.5	40
2	2.7	63.0	40
3	1.8	52	20
4	2.2	43	20
5	2.2	31	10
6	3.5	18	1

Table 5: Axle Spacings for reject classification

axle spacing	min space	max space	number of bins
1	4.0	30.5	10
2	2.7	63.0	1
3	1.8	52	1
4	2.2	43	1
5	2.2	31	5
6	3.5	18	3

In this technique, hyper-bins of parameter space are occupied during training and then accessed during testing. The data is divided into three parts by random choice: org.trn, org.tes, and org.tst, the training and two testing sets. Of the approximately 13,510 vehicles in file org.tes and 13,649 vehicles in the org.tst, 282 and 355 vehicles, respectively, are rejected as not being in any of the

Table 6: org.tes data of georgia 1993 study probabilistic neural net 40 40 20 20 10 1 bins

truth\guess	1	2	3	4	5	6	7	8	9	10	11	12	rej
1	17	0.5	0.5										0
2		4703	1548		7								3
3	0.7	1579	2214	3	113	1		6					55
4				23	16	6							3
5		7	127	11	203			3					34
6				2		208			2				12
7									1				1
8			7		2	1		143					45
9						3			2146				146
10										4			18
11											49		11
12												12	4
13													0

Table 7: org.tst data of Georgia 1993 study

truth\guess	1	2	3	4	5	6	7	8	9	10	11	12	rej
1	18	1	1										1
2		4794	1611		7								4
3	2	1584	2153	3	116	1		7					42
4			2	13	14	3							2
5		7	118	13	206			2					31
6				2		221			5				18
7													1
8			10		3			134					45
9						2			2164				165
10										1			26
11											54		14
12												14	6
13													0

Table 8: Truth Guess Reclassification with network applied to org.tst with spacings from Table 5. If all the vehicles listed as rejected in Table 7 are reclassified with this net, less than one vehicle(0.4) is rejected. The 1 in truth item 10 under rej column is the count in Class 13.

t\g	1	2	3	4	5	6	7	8	9	10	11	12	rej
1	3.4	11.6	3.0										0
2	0.6	4458	1778	0.81	18.7	1.5	0.	3.16			.03		0
3	0.6	2040	1762	1.9	81	17.4	.4	67.7	.9		.7		0
4			0.6	23	17.5	7.8		.07					0
5		30.8	130	13.6	175	4.1	0.	23.4	8.6		.56		0
6		0.8	18.7	5	4.9	175		14	5.8		.13		0
7			1										0
8		3.3	62	.65	16.7	14.1	.5	100			1.1		0
9			3		4.7	6			2212		69.4		0
10					.5					20.1			(.4) 1
11			.5		.2	.1		.5	55.6		3.1		0
12						1.3						14.6	0
13													0

bins established during training. A methodology that we plan to adopt in the near future is to examine each of the rejected vehicles and determine how close they are to the established data set. Then to incorporate them if appropriate. The reclassification neural network classification is shown in Table 8. Of the combined nets, only one vehicle is unclassified. The rejected Class 10 and Class 12 vehicles are shown in Table 9. These numbers are consistent with Tables 5 and 8.

The classification results are shown in Tables 6 and 7. These are seen to provide a much cleaner division in Class 2 and 3 and Class 3 and 5 than provided by Scheme F. Here the errors of classification of Class 2 as Class 3 and Class 3 as Class 2 balance out, giving the correct overall numbers in each class. Likewise, the errors in identifying Class 3 as Class 5 and Class 5 as Class 3 cancel out in a similar fashion.

Table 9: Hyper-Bins of Rejected vehicles in Class 10 and Class 12

as1	*as2*	*as3*	*as4*	*as5*	*as6*
class 10					
24	2	5	2	1	0
9	1	1	6	1	1
13	2	12	2	1	0
12	2	3	1	1	0
9	2	9	2	1	0
13	2	9	2	1	0
19	2	13	2	2	1
9	1	1	6	1	1
13	13	4	7	1	0
9	25	1	1	1	0
9	26	1	1	1	0
9	25	1	1	1	0
16	2	2	20	1	0
15	2	3	1	10	1
class 12					
11	2	6	7	5	0
21	2	8	4	8	0

3. Utility Theory

When there are many applications for a given set of data, utility theory can help sort out the important aspects of the data. The details of a particular application govern the utility function. One can imagine many scenarios of usage of traffic data: the most common involves estimation of a parameter at a particular site that depends linearly on the number of vehicles in each class, or a derived quantity such as the average speed times the number of vehicles, or perhaps, the equivalent single axle loadings of all vehicles. An enforcement utility function, on the other hand, might involve specification of the characteristics of a single stolen vehicle. The best classification scheme depends on the characteristics of the associated utility function. For estimation the following utility function is appropriate for repeated samples with the same distribution at a given site:

$$Q = -\sum_{ij} W_i (M_{ij} - M_{ji})^2$$

where W_i indicates the importance of a particular class and M_{ij} is the performance matrix element. This indicates that cancellation of errors is useful,; thus, if Class 2 vehicles are misidentified as Class 3 in the same number as Class 3 vehicles are misidentified as Class 2, the overall counts will be preserved and to the extent that cancellation happens, the prediction with cancellation will be as good as that without.

The problem is that most classification schemes are not geared for one site. Even if they were, the variations in traffic flow at a single site can change with weather, degree of urbanization, technological factors, and the mission of society. In the language of neural networks, this phenomena is called the problem of generalization. The network is trained under one set of circumstances and is used for another; how does its performance change? In the case that the performance matrix showed perfect performance, this would be expected to hold at all sites. When cancellation of errors occurs, the cancellation effect depends on the volume of traffic in the two cancelling classes. This is shown in the simple model moving demarcation line model, shown if Figure 4.

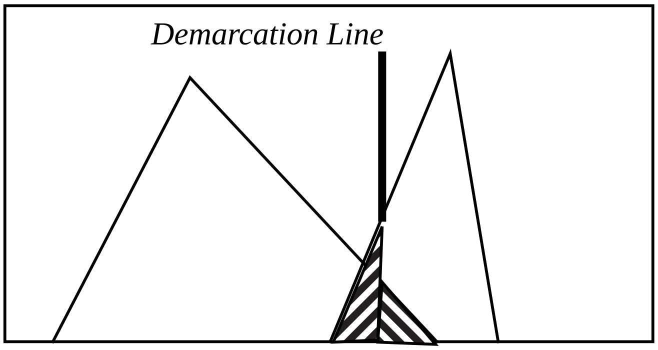


Figure 4: Moving demarcation line for intermediate case

The two distributions of vehicles are assumed to maintain the same shape, but have variable amplitudes. If there are almost no vehicles in the second class, the height of the second triangle will be very small. There will be almost no overlap area under it. This means that the demarcation line moves to the right until the area under the first triangle, to the right of the line, is almost zero. Contrariwise, if there are almost no vehicles in the first class, but lots of vehicles in the second class, the first triangle will have nearly zero height while the second triangle remains tall. Then the demarcation line moves to the left, until the area to the left of the demarcation line of the second triangle is nearly zero. An intermediate case is shown in Figure 4. Other models that vary with vehicle concentrations also show these effects and could be more important than the moving demarcation line model in practice. One such model would be to utilize a least square fit to the observed profile with variable coefficients multiplying two basis distributions. The conclusion is the same: namely a model’s generalization characteristics need to be examined in detail.

The generalization characteristics drives the importance of the diagonal matrix elements. The two effects of cancellation of errors and large diagonal matrix elements in the performance matrix suggest the following form of utility function to decide on the best classification scheme with data derived from one site to be applied at other sites for the purpose of parameter estimation.

$$Q = \sum_i W_i \left[M_{ii}^2 - c \sum_j \left(M_{ij} - M_{ji} \right)^2 \right]$$

Here W_i indicates the relative importance of class i. The parameter c is chosen large if the same distributions of vehicles occur at all sites and small if the variations at sites are totally unpredictable.

Conclusion

The undesirable characteristics of Scheme F have been described qualitatively and more generally with utility theory. A recurrent probabilistic neural network with a reclassification network for the rejected vehicles has been constructed and is a major remedy for the difficulties of Scheme F. The framework of utility theory can be the basis for determination of a best algorithm that applies to a variety of sites. With this and a study of the underlying phenomena at various sites, the details of best algorithm implementation can be determined.

References:

John H. Wyman, Gary A. Braley, Robert I. Stevens, Field Evaluation of FHWA Vehicle Classification Categories - MDOT, 1984, Maine Department of Transportation, Bangor, Maine, 04401.

Bruce A. Harvey, Glenn H. Champion, Steven M. Richie, Craig D. Ruby, Accuracy of Traffic Monitoring Equipment, GDOT 9210, Technical Report GTRI Project A-9291, June 1995.

W. S. McCulloch and W. A. Pitts, A Logical Calculus of the Ideas Immanent in Nervous Activity, Bull. Math. Biophysics 5 115-133, 1943.

Robert Sedgewick, Algorithms in C++, Addison-Wesley Publishing Company, Reading Massachusetts, 1992.

Acknowledgments:

The authors acknowledge the support of the FHWA, the helpful assistance of the Alliance for Transportation Research, and the assistance of Larry Blair at Los Alamos National Laboratory in securing funding under the auspices of the Department of Energy. The helpfulness of the management and support personnel of groups XCM and EES-5 at Los Alamos National Laboratory facilitated the accomplishment of this work. The technical content was influenced by helpful discussions with Drs. Rajendra B. Patil and Azim Eskandarian.

Detection of Catastrophic-Fault-Prone Classification Data

C. James Elliott

Group EES-5 Los Alamos National Laboratory

cje@lanl.gov

Abstract

Malfunction of commercial traffic classification equipment is a severe problem for the Federal Highway Administration (FHWA) that collects and disseminates highway statistics. The nature of the faults is such that a majority fall outside the bounds of fault-free ground truth data and are, therefore, said to be *catastrophic*. A neural network is used to generalize the ground-truth classification in a tractable manner. A second simpler go/no-go method is also used to clarify the fault analysis principles. We show that the majority of these faults can be detected without ground truth using a recently described sparse probability neural network algorithm accommodating hypervolume fuzzy interpolation for generalization (SPNN/HFI). The go/no-go method which also detects the faults is substantially less accurate. We further examine the temporal characteristics of the data faults of equipment of three vendors. These data were taken near Atlanta Georgia on I20 in 1993. In network terminology, these faults are transient and Byzantine because they are of short duration and they alter the output axle separation measurements from their true value. The catastrophic (observable) faults tend to occur for 18 wheelers while the faults are contained for two axle vehicles masquerading as other vehicle types. This analysis raises the question of the origin of the faults and offers a means that can become an instantaneous method of detecting them numerically. With immediate diagnosis, observation of faults under a variety of conditions can aid in pinpointing their cause. In summary, a numerical fault indicator for axle-separation based traffic monitoring equipment appears quite feasible. It can serve as a quality measure and an investigative tool to track down the causes of the intermittent behavior. This work is an example where the Vapnik-Chervonenkis distance is an inappropriate measure of capacity for dichotomies having known measurement errors or other mechanisms that guarantee a smoothness of the data.

1. Introduction

Fault detection is a specific application of neural networks and appears in many forms. Fault detection in the absence of pinpointing the origin of the fault is simply a dichotomy[1]. When multiple categories of faults occur with or without pinpointing, the problem is simply that of vector classification, by a multiple output neural network. Gertler[2] and Frank[3] reviews complexity and dynamical kinds of model-based fault detection where faults indicate deviations from expected model-based behavior. In a totally different field, Sadohara and Haraguchi[4] detect faults occurring in the process of synthesis of logic programs by example. Almost any kind of neural network can serve as a dichotomy. The *No Free Lunch Theorem* of David Wolpert [5] suggests that no neural network is ideally suited to every problem. A single layer perceptron [6] is known to be well suited to convex dichotomies. Backpropagation used with multi-layered perceptrons can handle disjoint regions, but we have no guarantee of converging to unique solutions with this technique. Cascade correlation[7] appears to be more general and could be combined with backpropagation units rather than single neurons. Top-down classifiers such as cluster algorithms are also suitable. Radial basis functions and other Parzen window approaches are known to be faster than backpropagation[8] but these have not been characterized as having tractable training and testing. Recently, a method called CMAC has been extended with cell-limited basis functions[9].

The problem to which we specialize is that of determining a large number of fault events of a particular variety in a large data set. A fault event is a measurement that does not match up with an independent and highly accurate determination called *the ground truth*. Pelc [10] describes five network terms for fault conditions that are relevant: (1) *permanent*, or (2) *transient*; (3) *crash fault,* or (4) *Byzantine fault*, and (5) *fault prone.* Permanent indicates a condition during the entire data collection of consideration during which it is useless to transmit data. This condition is known to occur during gathering of traffic monitoring data[11]. *Transient* indicates a non-permanent fault; the word suggest a brief period, existence, or sojourn, but we do not distinguish between *transient* and *intermittent*. A crash fault occurs when the entire communication system goes down and no data is transferred. On the other hand, a *Byzantine fault* describes arbitrary (even malicious behavior), it can include stopping, rerouting, or altering messages. *Fault prone* is used to indicate a significant fraction of events that hounds the usefulness of the data; in no way does it imply faulty equipment, equipment which in principle might be adjusted by a change in a few parameters and be rid of the latent condition.

When examining a new test data set without corresponding ground truth it is impossible to determine if a data point that lies outside of the generalization of the ground truth data is a fault or is a rare (true) event. A point within the category of c*atastrophic* faults will do violence to the ground truth location of the point in input space placing it in *terra incognito* a substantial fraction of the time, outside of the range of the ground truth generalization. If the system is fault prone, although events can not be distinguished individually, then on average the rare events will by comparison be small and negligible. Under these conditions, a fault detection system can be constructed without additional on-line ground truth. A *significant number of fault events* simply means that in a portion of parameter space the fault-event count is well above the rare-event count. The definition of a *rare event* can also include low probability events within the ground truth training set itself, and this type of consideration could be important if a large number of moderate faults or *peccadil-*

los occur which are just within the boundaries of the ground truth generalization and are of higher density near its boundary than the ground truth events themselves. The other variety of faults is *contained* faults that are defined to be those that leave the system state vector within the generalization of the ground truth data. These events are undetectable in a dynamical-free system unless the ground truth is available.

We specifically analyze faults from a set of data used for classification of vehicles into the 13 classes (see below) of the FHWA accompanied by the ground truth. It has been well known[11] that transient classification performance plagues commercial classification equipment. Although the cause or causes of the transient behavior is not known, it is believed to be associated with electrical or mechanical properties of sensors and/or distributed processors subject to free or bound electrical or mechanical noise. Mead[12] has also suggested environmental changes leading to electrical drift faults could be responsible.

We aim to construct a robust numerical algorithm to identify when a specific class of commercial data acquisition and processing equipment malfunctions. The equipment is used to obtain measurements of the number and separations of axles of passenger and commercial vehicles, whose tires apply impulses of stress to piezo-electric sensors embedded in the highway. Six or less such axle-separation distances constitute the input to the neural network for each vehicle. The ground truth measurements of the axle (tire tangent to the highway) separations are made from video images of the vehicles [13] and are compared to data sets provided by the manufactures[11] who participated in the experiment. The data streams are matched by a number of techniques that rely primarily of the time stamp of the events[13]. The actual data sets also contain classification and speed information of the vehicle as determined by the manufacturer of the equipment and the ground truth classification but these aspects were not a part of the current work, except for noting catastrophic speed errors in one case, and are described elsewhere[14].

Work by GTRI students and faculty led to what proved to be a high-quality video ground truth data set. This data set contained axle-separation and time-stamp data that corresponds to vendor measurements of the same quantities. The two sets were collated and analyzed for correctness. The GTRI final report points out a variety of errors including dropped axles, split vehicles, fused vehicles. Tasks that included checking for effects of temperature variation and weather yielded no positive correlations. Prior to this study, we made[14] a new collation algorithm and designed a cascaded SPPN/HFI neural network. We also made several hundred corrections to the ground truth, involving well under 1% of the data. We produced a neural network that could classify all 13 classes. It also was used to improve parameters in a model-based expert classifier known as the AC-CODE algorithm[15]. But none of this work addressed the question of how vendor equipment with intermittent faults could be detected numerically in a way that could ultimately be incorporated in an on-line fashion.

In this paper we describe the application of a GO/NO-GO and a SPPN/HFI classifier to the fault diagnosis problem. We compare it to what proves to be a less accurate but easier to understand algorithm that places go/no-go limits on each of the possible 6 axle separations. The Vapnik-Chervonenkis distance has been taken as the traditional measure of the capability of such a dichotomy. We compute the Vapnik-Chervonenkis (VC) distance [1] for each of these methods, their label distance (described below), and compare their diagnostic capabilities in determining the fault condition.

2. The Approach

The first method is a classical quality control method typically used with two mechanical gaps. The mechanical object must go through the larger gap but not go through the smaller. The width of the object has upper and lower bounds. In each variable of our network, space is divided up into three disjoint regions: (1) below the minimum of the ground truth data (2) between the minimum and the maximum of the ground truth data, and (3) above the maximum of the ground truth data. The no-go regions are (1) and (3) and the go region is (2). The go/no-go analysis is used for each dimension. That is, if and only if one or more of the independent axle spacings gives a no-go result, the entire result is no-go and it is said to be a fault. The Vapnik-Chervonenkis distance (VCD) for p inputs is computed. This distance is regarded as a capacity measure of a dichotomy; it is the number of training examples that can be guaranteed to be learnable without error irrespective of their class or exemplar location. We also compute the label distance which is a measure of the specific number of dichotomies producible. In this case it depends on the number of bits in the computer floating point representation and is not directly relevant to capacity.

Using the SPNN/HFI technique, there are N regions partitioned into bins for each of the inputs. Depending on the training data, each bin can take the value of either GO (G) or FAULT (N). The HFI generalization is a conservative technique for fault detection. When a fault falls within a non-boundary half-cell, the cell and the adjacent cell are taken to be in the fault zone. For interior points in p dimensions this fills 2^p cells for each exemplar. Here we also compute, in a closely related problem, the VCD and LD in p dimensions. The algorithm is applied to all classes grouped together and is also diagnosed by class. We show typical results for one vendor although three vendors were studied.

The adequacy of the technique we use is dependent on their being sufficient catastrophic faults and sufficient training data. Improvements on the technique requires additional a priori knowledge. Cluster algorithms with pruning is another approach that is believed to be adequate; however, the details of the pruning need to be tuned to the requirements of the problem. Likewise, many other neural network techniques could be adapted to this type of problem.

3. The Results

We modified the SPNN algorithm for the purpose of delimiting the space of ground truth, fault-free data. We considered any positive probability as indicating data occupancy of ground truth. After training on the ground truth (video) data, we ran the algorithm on vendor data. We also computed the corresponding VCD and LD distances. The six axle-spacing variable ranges are: [4,30.5], [2.7,63], [1.8,52], [2.2,43], [2.2,31], [3.5,18], with values in 30.48 cm units (feet). In the SPNN approach the cell numbers are p1=40, p2=40, p3=20, p4=20, p5=10, and p6=1 from which there are p1*p2*p3*p4*p5*p6 logical hypercubes.

Fig.1 Hourly vehicle count

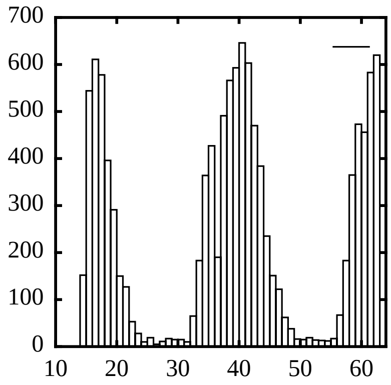


Fig. 2 NN rejects for all classes

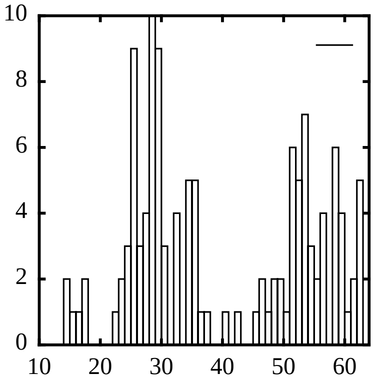


Fig. 3 Separation out of bounds

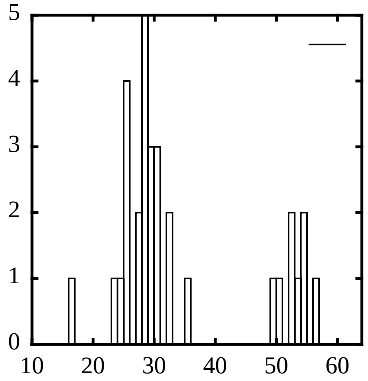


Fig. 4 Ground truth difference

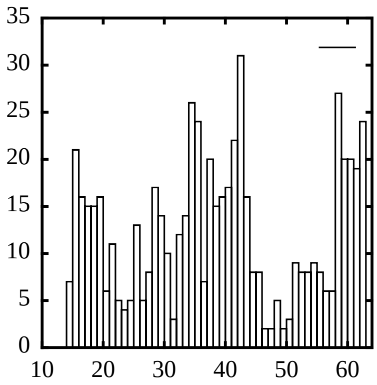
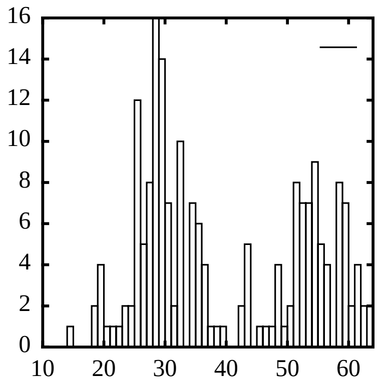


Fig. 5 Class 9 differences only



Our classification results were computed on a SUN/IPC computer and our asymptotic analysis was performed with Mathematica on a networked machine. The quantities plotted are by hour of the day over the utilized portion of the three day period. The hourly count in Fig.1 shows some correlation with the ground truth differences of Fig. 4, but the Class 9 portion of these shown in Fig. 5 appear uncorrelated. The neural network (NN) rejects (faults) are defined to occur when the vendor classification using a method called Scheme F differs from the ground truth classification using the same technique. This method disregards any vendor provided classification and utilizes the measurement of the axle spacings only.

In the appendix, the VCD distances were found to be 2 or 3 in both cases, but using an easier and slightly modified model for the SPNN. The LD distances were found to be related to machine precision for the GO/NO-GO analysis. The LD distances were related to the number of cells in the SPPN method, lying between 0.5 bits/cell and 1.0 bits/cell. The asymptotic LD for many cells is 0.81 bits/cell.

4. Discussion

The study was interesting from the point of view of demonstrating the existence of fault-prone data and that the VCD does not seem to be a measure of the capacity for the SPNN method. The results show that Class 9 vehicles are those that give fault-prone conditions and that in principle a fault-prone indicator could be very useful in detecting the cause of conditions similar to those of the Georgia I20 study. The use of Scheme F was shown in a previous study to be highly accurate on Class 9, approximately 99% accurate. The reason the neural network has rejects (faults) that do not track the ground truth differences is that some of the faults are of the contained variety. Indeed most of the two-axle vehicle faults map outside their class but fall into another two-axle vehicle class. Such contained faults are not observable by vendor data only. We also see that the SPNN technique is more powerful than the GO/NO-GO analysis; for instance, the SPNN technique appears to pick up a small amount of the non-Class 9 vehicles.

The values of 2 or 3 for the VCD do not represent the concept of capacity of the SPNN network. The VCD is related to convergence to a true probability distribution. If one has a distribution that falls off rapidly to zero in a fraction of a cell size, the generalization of SPNN will displace part of the distribution a cell over the edge of the cell and therefore the network converges to the wrong value. This is a reason for its small VCD. However if we know *a priori* that the probability distribution will be smoothly varying over a cell size, it is clear that the VCD does not measure the capacity of the network for SPNN; rather the label distance which counts representable patterns, appears to be a better measure. The label distance is not appropriate, however, for the GO/NO-GO method. A modification of the VCD that forbids the choice of new test points to be within a fixed box centered on previous test points would bring the VCD into line with these notions of capacity if the box were chosen to be close to the cell size or local smoothness cell size. To lowest order, the idea is that within the box all points are known to have the same class; why require test points within the box that are known not to occur? An alternate but peculiar way to accomplish the modification without substantially changing the concept of VCD would be to restrict the space of test points for VCD computation to be on a lattice of cell centers. The real test data could not be handled in that manner if we wished to maintain generalization.

The asymptotic decrease from unity to 0.89 bits/cell of the LD in SPNN is associated with the increase in generalization, that is, a training or testing data point is assigned not to just one cell but also activates a number of neighboring cells. Any such neighboring cell is treated the same as if it were activated by a data point directly falling in the cell. This generalization capability mimics experimental error associated with making the measurements in the first place. It also is important in smoothing the data were insufficient data is available.

We believe these results to be new to the transportation community and should be a stimulus for further diagnosing and determining the cause of Byzantine faults. The computation and discussion of the VCD has shown it to be an inappropriate measure of capacity for dichotomies of the SPNN variety.

5. Conclusions and summary

We have demonstrated two means of diagnosing the 18 wheeler subset of vehicle faults for one vendor (typical of the three vendors studied). Of the two techniques, the SPNN is more accurate but is more difficult to implement. Either of these techniques would be useful as an on-line diagnosis to help further study the origin of these problems. We also conclude that the Vapnik-Chervonenkis distance in not an appropriate measure of capacity of such a dichotomy when it is known that data are smooth from measurement errors or other causes.

6. References

[1] Haykin, S., **Neural Networks**, 80-83, Prentice Hall, Upper Saddle River, New Jersey
[2] Gertler, IEEE Control System Magazine **8**, #6, 3-11
[3] Frank, Automatica **26** #3 459-474
[4] Sadohara and Haraguchi, Lecture Notes in Artificial Intelligence **997**, 266-281
[5] Wolpert, David H., The mathematics of generalization, **The Proceedings of the SFI/CNLS Workshop on Formal Approaches to Supervised Learning**, ed. David H. Wolpert, 1992, Addison-Wesley 1994, Reading MA [not obtained]
[6] M. L. Minsky, S. A. Papert, Perceptrons, Expanded Edition, 1988, MIT Press, Cambridge MA
[7] Haykin, Op. Cit. p206 see: S. E. Fahlman and C. Lebiere, **Advances in Neural Information Processing Systems 2** (D. S. Touretzky, ed.) 524-532, San Mateo CA: Morgen Kaufmann
[8] Sprecht, Donald Probabilistic Neural Networks and general Regression Neural Networks in **Fuzzy Logic and Neural Network Handbook**, ed. C.H. Chen, McGraw-Hill Series on Computer Engineering
[9] Ching-Tsan Chiang and Chun-Shin Lin, **Neural Networks 9** #7 1199-1211
[10]Andrzej Pelc, **Networks 28** 143-156 (1996)
[11]Harvey, Bruce. A., Champion, Glenn H., Ritchie, Steven M., Ruby, Craig D., **Accuracy of Traffic Monitoring Equipment**, June 1995, Technical Report GRTI Project A-9291 (9210).
[12]W. C. Mead, private communication
[13]R. Gillmann, C. J. Elliott, J. Pepin, *Super Synchronization for Fused Video and Time-Series Neural Network Training*, **Neural Network Applications in Highway and Vehicle Engineering Conference Proceedings**, April 10 -11, 1996, George Washington University, Ashburn Virginia
[14] Elliott, C. J., Pepin, Jason, Gillmann, Ralph, **Proceedings National Traffic Data Acquisition Conference**, Albuquerque, NM, May 5-9, 1996., Vol II, 17-31.
[15] **Traffic Monitoring Progress,** 5-11, July 1996.

7. Acknowledgments

Ralph Gillmann of Federal Highway Administration for contract oversight; Tom Caudell for organization suggestions and persistence in demanding a manuscript; Judith Binstock for suggesting taking Tom Caudell’s class; James C. Scovel for pointing out the Wolpert reference; computing environment support by Lynn McDonald and Anthony Montoya; FHWA for financial support; Francis Knudson for selecting new books at the Laboratory library; my family for time that could have been theirs; any pertinent authors whose work I have seen and forgotten; and, above all, an individual who wishes to remain anonymous who suggested the spirit of the work and arranged many of the requisite meetings.

Appendix: The VCD analysis and the label dimension of the GO/NO-GO and SPNN/HFI

We analyze the Vapnik-Chervonenkis and label dimensions for the GO/NO-GO classifier and a slight modification of the SPNN/HFI classifier.

First consider the GO/NO-GO case. The VC distance is with respect to a distinct set of divisions of space. The number of distinct ways the regions can be divided depends on what is usually called the training parameters or weights. These parameters here are twofold, the minimum and maximum of the GO range. The learning machine encompasses all possible choices for these parameters. Here it is clear that the number of ways or dichotomies into which space can be divided depends on the characteristics of the word size or precision of the computer itself. That is suppose F= 2^f is the number of ways a floating point number can be represented in a machine. Then the first minimum can be chosen in F-1 ways. For large F, then the pair of minimum and maximum can be chosen roughly $F^2/2$ ways. This means that the 1-D problem has 2f-1 bits that designate or label dichotomies; the label dimension LD is then equal 2f-1. Not all of these dichotomies are useful in being able to shatter a set L of a number of |L| points. To shatter the set L, there must be $2^{|L|}$ distinct training modes out of the 2^{2f-1} possible labels. With shattering, all possible independent assignments of GO (G) and NO-GO (N) to each of the L points must have a corresponding dichotomy label, irrespective of where the L points are selected within the p dimensional vector space X. With G/N the two free parameters are sufficient to accommodate one or two points. However for three points with the assignments GNG, respectively, the NGN structure of permitted labeled dichotomies can not be replicated. Thus the VC distance is two for the 1-D G/N machine. For p independent inputs, the VC dimension is at least as large as 2, because then the classification can always occur simultaneously in a p directions. It also cannot be greater than 4 because all of the test points can be required to cluster in all dimensions but x. One additional point is then specified that is far away in all dimensions except x in a positive sense. The x component of the additional point is chosen the same as another x component and assigned the same truth value G or N as that point. The outlier point is assigned N in all dimensions except x where it is assigned G. The cluster points are assigned G in all dimensions but x. By the addition of one point, the problem is reduced to the 1-D problem on the original points.

The VC distance in the SPNN/HFI machine is certainly greater than or equal to unity. To make the problem slightly easier for these estimates and without loss of fundamental ideas, the grid will be assumed to be periodic, as if the low and high boundaries were adjacent to each other. The grid can either be uniformly magnified or translated and still be a SPNN/HFI machine. Any point with either VALID (G) or FAULT(N) designation can be shattered by choosing the point to be inside any cell. But consider the case of two data points. When the two points occur on top of each other and one has classification N and the other G, no dichotomy within the family can shatter this case. Haykin's treatment of shattering does not explicitly eliminate this case, a point that could have been made earlier. We assume here that the fix to Haykin's treatment is that we can eliminate any number of special cases such that they have no volume associated with them. The set of dichotomies are those made up of a grid of any length having a fixed number of interior partitions, C, and no points allowed beyond those boundaries. It may be thought of as magnifying ruler or graph paper with C+1 marks. With 2 input points, it is clear that the boundary of the ruler can be moved and fractionally magnified to have two divisions between the two points. (The reason for two divisions is to allow for generalization that places two points in the grid for every training point). This is always allowable with the number of cells, C >= 3. With three points, the fractional magnification and displacement can be arranged to accommodate the two closest points, x1 and x2, but then requires sufficient cells to accommodate x3 should it be far from x2. With C fixed and x3 free to be anywhere, there is no way to shatter the data with this cardinality of three points. Thus the VC distance is 2, the same as that for the GO/NO-GO machine. With p inputs, LD is at most 3. This may be proven similar to the previous GO/NO-GO machine. The test points are chosen to cluster in all dimensions but x. An outlier is chosen as before but with the additional requirement that it is far enough away in each direction except x that all the cluster points lie within the same half cell for those directions. Again the problem is reduced to a 1-D problem by the addition of one point.

The label distance is just as easy to compute for SPNN/HFI without the boundary modification as it is with the modification. We will address the unmodified enumerations. The learning machine as modified doesn’t permit patterns that start off as GN and end with G whereas the unmodified machine does not exclude these cases. The label dimension can be determined by enumeration of the possibilities with the following rules for B bins in one dimension:
(1) The first, b=1, and last, b=B, bins can be either FAULT (0) or GO (1) independently of any other data.
(2) The cells b=2 to b=B-1 may be occupied by any number of runs of 0’s (no data) of any length or any number of runs of 1’s of length 2 or more providing the binary digits fill the B-2 locations.

In the special case of B=2, no internal state can be occupied, and 2B states occur.
In the case of B=3, the 4 external states are 000, 001, 100, 101, where no data is in the internal state. Of these all but the first can have the middle bit changed to unity. Thus the number of possible states is 7 or $2^{0.936B}$.

Suppose we know the number of each of three types of patterns of length N, labeled p,q,r corresponding to ending pairs of 0x 10 and 11, where x is 0 or x is1. If we extend N by 1 on the beginning side, the N+1 possibilities are P,Q,R. Invoking the rules, these are related to p,q,r, by the following relationships: P = p+q; Q=p; R=r+q or

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}^N \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

after N applications of this rule: This formulation allows one then to compute the VC dimension by using (p,q,r)= (2,1,1) at N = 2 as the starting right hand side vector. The final vector of values is consistent with these same rules giving a total number of patterns as $P+Q+R=2^{LD}$;where N= number of cells and LD is the label distance. It should be clear that every increase in N by two cells adds at least one free binary degree of freedom, and if all the patterns eliminated by the rules were replaced, an increase in N by two adds two free bits. Thus, the LD dimension falls between N and N/2 for this case, the exact form being the log to the base 2 of the sum of three exponentials (the largest of which is 1.267 exp 0.5624N) is too complex to give much insight, but the asymptotic form for large N is 0.811372 bits per cell. With 6 cells the result is 0.87 bits per cell. Asymptotically, for large numbers of cells, the enumerations in each dimension are independent of the enumerations in the other directions and are irrespective of the boundary conditions. In multidimensional spaces, then, the same limits on the LD dimension and the same asymptotic form applies. The linear generalization achieved by the interpolation drops the LD dimension by 0.19 bits/cell. It is conjectured that one always trades LD dimension for generalization. The loss in LD dimension does not necessarily constitute a measure of the performance of the generalization, however, because LD dimension can be thrown away by a machine which for instance only stores information in even cells and ignores the odd cells.

A Tractable Sparse Probability Neural Network

C. James Elliott

Group EES-5, Los Alamos National Laboratory

cje@lanl.gov

Abstract

We describe a neural network algorithm for constructing tractable sparse probabilistic neural networks (T-SPNN) suitable to sparse occupancy of the input space having P dimensions. Sparse occupancy indicates that as P increases, the training/testing and volume of the lattice, L, remains at most polynomial in the number of dimensions. The number of sparse points is the lattice training volume. Similar to the Albus CMAC algorithm, it fits the Werbos paradigm of syncretism, and is similar to area sampling and stratified sampling. Like CMAC, SPNN is hash-table based. The algorithm incorporates a concept similar to the CMAC shifted receptive fields by using fuzzy weights. Limiting the number of fuzzy weights by taking the C, a number which is fixed or at most polynomial in P, most important ones, the algorithm exhibits tractable (polynomial time) behavior for polynomial bounded training and testing, in contrast to already established intractable exponential time for perceptrons and backpropagation. This method of generalization is called C nearest cell neighbors (CNCN). Taking C to be P+1, and using the P+1 nearest points in a grid file cell, leads to O(P) operations for collisionless fuzzy weight computation of each of the training points. The hash table operations vary from O(P) to O(PL), the former with few collisions as with low fill the latter with a one key hash table. With C > P, there can be generalization in all directions of the space. The emphasis is to make the algorithm general but understandable to a wide audience.

Introduction

The general area we address is that of probabilistic neural networks (PNN) which have been recently reviewed by Specht[1]. These have been constructed by Parzen window approaches such as radial basis functions which are known to be substantially faster than backpropagation. Smolensky[2] reviews complexity analysis of backpropagation and perceptrons showing that these algorithms require time that is exponential in the input space dimension, P, and therefore are intractable. Although the number of Parzen basis functions can be reduced by adaptive techniques where widths and centers can vary, the analysis of such an approach is difficult and Specht reports no results for convergence of such systems.

We specialize the PNN to the following fixed-center fixed-width problem. Given a P dimensional lattice grid with lattice spacing of unity and N_i points in each i'th direction and K classification types, we wish to construct a PNN that can describe probability variations of order unity in each step on the lattice in any arbitrary direction. A naive PNN on a lattice grid with N_i =N would then require N^P centers and the resultant computational work makes naive PNN intractable.

As it is not wick to address all possible remedies to intractability, we introduce one such algorithm we call a sparse probability neural network (SPNN) known as T-SPNN in its tractable form and describe its properties. This algorithm requires specification of a generalization scheme of which we consider two. The C nearest cell neighbor (CNCN) generalization scheme assigns probabilities to a subset of C lattice points constituting the hull of a finite P-dimensional volume. And the hypervolume fuzzy interpolation (HFI) generalization scheme that uses $C=2^P$. The sparseness refers to the requirement that the occupied lattice points be at most polynomial in P, whereas the total lattice points are exponential in P. The training (and class-free testing) data must also contain a polynomial bounded number of point-class pairs. This requirement automatically guarantees sparsity of the lattice.

The algorithm has closer ties with the CMAC algorithm[3-5] than the conventional Parzen window algorithm. SPANN/HFI can be viewed as an Albus CMAC[3,4] neural net with the C receptive fields replaced by a fuzzy algorithm with 2^P receptive fields. Our dependence of C on the dimensionality is uncharacteristic of extensions of CMAC work[5]. Although our two generalization methods and the variable-base vector key are believed to be new to PNN, the technique also follows the spirit of a practical implementation of the *syncretism paradigm*[6] coupled with fuzzy generalization. The method also can also be viewed as *area sampling* or *stratified sampling*[7] accompanied by fuzzy generalization.

The present approach is but one of many that can be taken. Additional literature searches of a myriad of references described in [2,5] are in order. The paper is also incomplete because of lost observation of another reference that also deals with the requirements on the input space for tractability. Furthermore, a new algorithm by Maass & Turan said[2] to train in time that is polynomial in P has not yet been examined for relevance to the present method. Hash grid file references[8] have not been examined either. The C++ implementation of CNCN has been started.

In this paper we next describe the approach, its merits and origins, and the algorithm details and timings. Then we tabulate the aggregate results in two tables in the results section. This is followed by a discussion section including opinions of merits, interpretation, comments, analysis, improvability, relationship to literature, and future research. The summary and conclusions are followed by references and acknowledgments.

The Sparse Probabilistic Neural Network Approach

The lattice space consists of orthogonal evenly-spaced scales. On training each point is assigned a hypercube region whose origin is found by integer truncation of the intra-cell point. Each training point is assigned weights on a cluster of C nearby lattice points. The sparse aspect of the approach is an implementation feature handled by a vector hash table in a fashion using linked lists as are used for row vectors of sparse matrices[9].

The method is fast in training and testing; it requires storage that can be considerably smaller than the amount of training data. Above all, the method is provably tractable for polynomial bound training and testing data. The details of how the algorithm uses a hash table to construct the generalized lattice points are described in the algorithm section.

The origins of the model are primarily via CMAC. The hypercube bins and hash table ideas were taken from the CMAC algorithm. The 2^P fuzzy key set [10] evolved as a method of choice for generalization after fuzzy discussions. It was then observed that intractability of the full fuzzy key set occurred because of the using of all the weights and that tractability could emerge by limiting the set to a finite or tractable number of weights. The definition of sparsity follows from the intimate tie between the size of the fuzzy key set and the computational complexity. Further details are in the acknowledgments. The new algorithm is as follows:

1. Choose x_{i0}<= min(x_i); N_i the number of cells in the i'th direction; dx_i >= max(x_i)/(N_i;-1); null the Information vectors; P is the number of input lines; Tr/Te is the number of training/testing inputs; T indicates either of these; K is the number of classes in which to classify; L is the total number of generalized lattice points; H is the hash table size; Z is the maximum number of collisions per key over all keys; C is the number of generalization points per training point. A training point consists of a P-dimensional point **x** and a classification j into one of {1,2,...K}.

2. Quantize the vector as **q** on a positive integer lattice $q_i = [Q_i]$, $Q_i =(x_i-x_{i0})/dx_i$, componentwise; [] truncates fractions; lattice vectors are ordered by their i'th component, starting with i=1; if the i'th components are different, the process is complete; if the i'th components are the same, the order is determined by the i+1'th components. If all components are the same the lattice points are equal.

3. Compute the vector key(**q**(**x**)) $=q_1 + q_2*(n_1 +q_3*(n_2+q_4*(n_3+...q_P*n_{P-1})))...)$ mod H; where H is a prime chosen to be the number of available slots in the hash table; overflow is avoided by inserting mod H at the termination of each close parenthesis on the right hand side thereby keeping all quantities on the mod-H ring, as similar to the function, hash()[9]. Primes may be generated by the *sieve of Eratosthenes*[9] for moderately sized hash tables, but here we assume they are looked up in a known table. The special case of $n_1 = 12$, $n_2 = 3$, $n_3 = 1760$, admits the interpretation of inches, feet, yards, and with n_4 of miles where the full key is the total distance measured in inches, and the key is the full key mod H.

4. Compute the full fuzzy key set by adding **d** = (d1,d2,...,d_p) to **q** an element of the P dimensional positive lattice $I^P_{>=0}$. Choose the fuzzy key set having C values of **d** by HFI, CNCN, etc.

For HFI $d_1+2d_2+4d_3+8d_4+2^4d_5+...2^{P-1}d_P$ ranges from 0 to 2^P-1 and each d_i is positive and less than or equal to unity, i.e. **d** is an element of $\{0,1\}^P$, and the fuzzy key set is **q**+$\{0,1\}^P$, with each element a member of $I^P_{>=0}$; the weight for the j'th element of the key set is $w_j = W_1W_2W_3...W_P$ where for a particular **d** componentwise, $W_i = 1- d_i +(2d_i-1)*(Q_i-q_i)$, i = 1,2,...P.

For CNCN weighting, the cell origin is chosen in step 2 above. The hypercube hull of the vertices generated by extending the lattice by +1 in each of the P directions constitutes the boundaries of the cell. Within the cell, a simplex is chosen having C vertices as the closest of the 2^P vertices of the hull to the intra-cell point **Q**. The vertices chosen are lattice points $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3,...,\mathbf{v}_C$, the key set. And the C equations $\mathbf{Q} = w_1 \mathbf{v}_1 + w_2 \mathbf{v}_2 + w_3 \mathbf{v}_3 ... + w_C\mathbf{v}_C$ with $w_1+w_2+w_3+...w_C =1$ are solved for **w**. Unless explicitly stated otherwise we take C=P+1 giving C unknowns.

5. Linked list work: Augment the key dictionary, adding a linked list node if required on training, for this ordered separate-chaining implementation of a linked list[9]. The k'th dictionary has the index function in(k,**q**) such that for colliding lattice points **q** and **q**' with k=key(**q**)=key(**q**); the ordering requires: in(k,**q**) = in(k,**q**') implies **q**=**q**'; in(k,**q**)>in(k,**q**') implies **q**>**q**'; and i(k,**q**) < in(k,**q**') implies **q**<**q**'. Each key, k, has a maximum index Imax(k). The index function does not have to be implemented; it is used to be explicit in specifying the algorithm. It says that all collisions, which have the same key, are linked in **q** order as described in 2 much like the addition of a new word to a dictionary in alphabetic order. The analogous index function gives the number of the word in the dictionary as a function of the word. Testing: search through the list for a match.

6. The i'th lattice point **q** (word) in key dictionary, k, has an attribute called Info for each class j = 1,2,...K. For each **q** in the key set, k =key(**q**) and i= in(k,**q**), increment Info(k,i,j) by w, the weight of the corresponding key set lattice point.

7. Normalize to probability estimators. Replace or augment the Info by a floating point number Info/N where N is set to zero and then incremented by Info(k,i,j), for j = 1,2,...K.

8. Return all K attributes of the i'th word of dictionary k averaged over the keyset words.
Train using 2,3,4,5,6 for each point-class pair; normalize using 7; test using 2,3,4, and 8.
Analysis of the operations(time) required for the computation are as follows by step in Table 1.

Table 1: Algorithm Step Complexity in SPNN Network

step	CNCN train	CNCN test	HFI train	HFI test	lattice prep
1	P	P	P	P	_
2	P Tr	P Te	P Tr	P Te	_
3	P Tr	P Te	P Tr	P Te	_
4	Tr P[Sort(C)+P]	TeP[Sort(C)+P]	Tr P 2^P	Te P 2^P	–
5	Z Tr K	TeZK / TeK log Z	Z Tr K	TeZK / TeK log Z	_
6	C Tr K	_	_	_	_
7	_	_	_	_	KL
8	_	C K Te	_	C K Te	_
standard use	C = P+1	C= P+1	C = 2^P	C = 2^P	–

CNCN method: Because of orthogonality of the basis, the weight solve time is bounded by proportionality to P. This gives a representation of **Q** as the weighted sum of the **v** quantities. Here w_i is the weight of the lattice point v_i of the key set. The total work for CNCN uses Sort(N), the operations to sort N items. Because Sort(C) > C, we need keep the term Tr *P*[Sort(C)+P] in step 4. Usually C=P+1.

In the collision free limit, the total time for link-list work will be proportional to the number of training points, Tr. When collisions abound, the total computational time will be bounded by proportionality to an additional factor Max(Imax(k)) over a subset of all the keys k = 0, 1,... H-1. This maximum over all keys is defined as Z. Hence the total link list creation work is bounded by proportionality to Z*T, in the collisionless limit we can take Z=1. For testing, the link-list retrieval time is also proportional to Z*Te, but using a binary search can be reduced to log(Z)*Te.

Results

We described the SPPN algorithm and obtained bounds for the number of operations in each stop required using the two generalization methods, HFI and CNCN. In CNCN none of the terms is intractable whereas in HFI generation of the weight set is intractable. Algorithm input requirements are spelled out in step 1. For testing, the probability estimate for each class is computed.

The aggregated results are tabulated below.

Table 2: Proportionate bounds for standard use numerical operations

generalization	Training	Testing	algorithm	tractable?
CNCN	TrP(K+Sort(P)) + TrK(Z+P)	Te(K+Z)	T-SPNN	Yes
HFI	$Tr2^{P}P$	Te(K+Z)	SPNN	No

Table 3: Bounds on Other Parameters using Poly for Polynomial Time

Parameter	meaning	collisionless	collisional
Z	collisions+1	1	<L<Tr
K	# classes	Poly(P)	Poly(P)
Tr	# training **x**	Poly(P)	Poly(P)
Te	# testing **x**	Poly(P)	Poly(P)
Sort(P)	sort time	Poly(P)	Poly(P)

Discussion

We have shown that T-SPNN is a highly effective algorithm for large neural network problems. Its learning and testing time using CNCN generalization and polynomial bounded classes and input data is polynomial in P. The intrinsic underlying fuzzy-free idea of the algorithm appears to be universal appearing in other guises in the neural network literature and as statistical techniques. The CMAC receptive fields could also be regarded as a fuzzy mechanism with fixed value of the parameter C, but with a different type of generalization in at most C-1 directions and requiring multiple training epochs compared to single epoch training of SPNN. The linear nature of the CNCN algorithm is a step towards fitting any patchwise linear function. If a patchwise linear function fitted has at most one patch per cell and that patch has sufficient training, CNCN is exact for testing in an isolated patch. Our discussion of SPANN has included all aspects of the algorithm in contrast to that of Judd[11] who for other algorithms describes only what he calls Loading.

Performance comments may be noted. Trading generalization for speed (C=1) can be accomplished by removing the fuzzy boundary. SPANN is much faster for testing than a radial basis function Parzen window having the same number of nodes as hash table lattice points, because SPANN evaluates only one cell rather than basis functions of all cells. For other types of hash tables see Sedgewick[9]. Using SPANN/ HFI gives good generality in each dimension but is slow. Using the CNCN method, we get generality in each dimension and tractable time with good coverage and good generalization. Detailed analysis of the computational complexity requires a choice of the sort routine. For high P, a large-radix sort in a fixed word machine can be linear in P and may be preferred over a quick sort that takes P log P operations on average or P^2 at worst[9].

Additional work is required. Non-linear preprocessing of the dimensions having non-uniform scale size could be of advantage. Computation of the accuracy of the data and returning an interval of accuracy could be helpful. Future directions of this work involve partitioning space into P-dimensional simplices rather than hypercubes and seeking training schemes requiring ~n log n time and testing schemes requiring average log n retrieval time where n is the number of training points. Work on multigrid, R-Trees[8], and median KD-trees[8,9] is relevant.

Summary and Conclusion

T-SPNN offers a needed, tractable algorithm, that should be applicable to a wide variety of problems involving large data sets especially large testing sets. When the smoothing scale varies, the algorithm can be used with the minimum scale size in each dimension as the cell size. For our uniform scale problem T-SPNN/CNCN is ideally suited and tractable.

The exponential learning in P for perceptron and backpropagation algorithms does not take advantage of the data sparsity. The Maass & Turan[11] algorithm exhibits polynomial loading, but its testing properties are not described. Single layer perceptrons cannot describe disjoint convex regions. Backpropagation algorithms suffer from multiple minima. Our solution requires only one epoch of training and is tractable both in testing and training. For two transportation applications[12,13] SPANN/HFI appears adequate, but the rigid grid structure could best be altered for other applications and the tractable version used when computation resources are limited.

In conclusion, SPANN/CNCN is a fast tractable neural-network algorithm for problems having tractable training and testing volumes and a tractable number of classes. In spaces with highly varying scale sizes it loses some of its appeal.

Acknowledgments

Ralph Gillmann of Federal Highway Administration for contract oversight; Tom Caudell for organization suggestions and persistence in demanding a manuscript; Judith Binstock for suggesting taking Tom Caudell’s class; Sara Matzner for expressing interest in this method and thereby focusing the presentation; Rajendra Patel for fuzzy logic discussions; Andrew Kuprat for advanced disclosure of his work on meshes and references; Richard Beckman for statistical references; Azim Eskandarian for an introduction to CMAC; Richard Lee Morse for an explanation of area weighting, the prototype for HFI; computing environment support by Lynn McDonald and Anthony Montoya; FHWA for financial support; Francis Knudson for selecting new books at the Laboratory library; my family for time that could have been theirs; any pertinent authors whose work I have seen and forgotten; and, above all, an individual who wishes to remain anonymous who suggested the spirit of the work and arranged many of the requisite meetings.

References

[1] Sprecht, Donald F., Section 3.1.7 in **Fuzzy Logic and Neural Network Handbook**, ed. C. H. Chen, McGraw-Hill, New York, 1996

[2]Smolensky, P., **Mathematical Perspectives on Neural Networks**, Ch. 2, *Overview: Computational Perspectives on Neural Networks,* P. Smolensky, M. C. Mozer, D. E. Rumelhart, editors, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 1996.

[3] Albus, James, **Theoretical and Experimental Aspects of a Cerebellar Model,** Ph.D. dissertation, University of Maryland, 1972 (not yet obtained)

[4] Albus, James, **Brains, Behavior, and Robotics**, Byte Books/McGraw-Hill, Peterborough, NH, 1981 (not yet obtained)

[5] Miller, W. Thomas, III, in **Fuzzy Logic and Neural Network Handbook**, ed. C. H. Chen, as **Cerebellar Model Arithmetic Computer**, McGraw-Hill, New York, 1996

[6]Werbos, P., *Supervised Learning: Can It Escape Its Local Minimum?*, in **Theoretical Advances in Neural Computation and Learning**, ed. by Vwani Roychowdhury, Kai-Yeung Siu, and Alon Orlitsky, Kluwer Academic Publishers, 1994.

[7]Cochran, William G., Recent Developments in Sampling Theory in the United States, Ch. 40 in his **Contributions to Statistics,** John Wiley and Sons, New York, 1982 from Proceedings of the International Statistical Institute, 3A, 40-66, 1947.

[8]Ooi, Beng Chin **Efficient Query Processing in Geographic Information Systems,**1990

[9] Sedgewick, R., **Algorithms in C++**, Addison-Wesley Publishing Company, Reading Massachusetts, 1962. Chapters 3, 16, 36 (C and Pascal versions are also available).

[10] Elliott, C. J., Pepin, Jason, Gillmann, Ralph, **Proceedings National Traffic Data Acquisition Conference**, Albuquerque, NM, May 5-9, 1996., Vol II, 17-31.

[11] Judd in **Mathematical Perspectives on Neural Networks**, Chapter 5, Op. Cit

[12] C. J. Elliott, Detection of Catastrophic-Fault-Prone Classification Data, this report

[13] C. J. Elliott, Beyond Scheme F, this report

Extracted Georgia-I20-Study Articles

The following articles have been extracted from progress reports from issues of Traffic Monitoring Progress (TMP) 6/95 to 12/96. They serve as a complete reference to TMP articles on the subject of the Georgia I20 Study, giving details of items mentioned in the In-Depth Review of the Georgia I20 Study. The articles appear in chronological order. Minor editing has occurred including renumbering figures and tables, providing uniform titles, and clarification of context.

6/95

Georgia Data Set Has Outlier Points

The ‘93 Georgia data set has been scrutinized for 2-axle to 6-axle vehicles. One vehicle record was found with a negative axle separation. Also, two single axle vehicles were found and removed from the set. Other data points appear different from most of the rest, and these are called outliers.

Ralph Gillmann of the FHWA has observed negative axle records for 9 axle vehicles and these have been confirmed at LANL.

The Georgia data set arrived from the FHWA in five zipped files called 5day.zip,orig2.zip, orig1.zip, nolnchn2.zip, nolnch1.zip. The nolnchn files are for those vehicles that remained in the same lane throughout the test.

We will utilize the files that include vehicles that make lane changes. These now have a total of 33,913 lines of data, one per vehicle record. However there is no new-line character at the end of the files.

The files broken down by axle and vehicles are:

25525	2 axle vehicles
1183	3 axle vehicles
779	4 axle vehicles
6297	5 axle vehicles
111	6 axle vehicles
(11)	7 axle vehicles
(1)	8 axle vehicle
(6)	9 axle vehicles.

Bruce Harvey of Georgia Tech Research Institute trained students to classify the vehicles based on the videos. They clicked a mouse at the various locations on the vehicle image to obtain the dimensions. No correction was made to the camcorder images for non-metric lensing. Corrections were made for lane position and parallax.

He believes that the one- axle vehicles were artifacts of the methodology in error correction situations. The negative lengths were attributed to equipment failure. The nine axle vehicles were very unusual and often military types.

The subclass is assigned to a vehicle only when it is pulling a trailer. In that case the subclass indicates the number of axles of the trailer.

A list of potential outlier points has been prepared and sent to Ralph Gillmann of FHWA.

7/95

Georgia Data Has Vehicles With Strange Axle Separations--The Outliers

The nine axle vehicles that Ralph Gillmann of the FHWA has observed to have negative wheelbase records and other vehicles observed at Los Alamos and also observed at FHWA are called outliers. These vehicles have strange properties. Some of them are strange because of data collection and processing errors. Others are strange because they are two or more vehicles components mated in unorthodox or unusual ways.

Thirty-seven video tapes of the ground truth of the Georgia data set arrived from the FHWA for examination of these outliers. One of the video tapes, which had been disabled because of the loss of the plastic knobs for hinges, was repaired temporally by a longitudinal adhesive tape along the bend.

Of the 25525 ***2 axle vehicles***, fifteen were identified as potential outliers. These fell in Classes 0, 2, 4 and 5. The Class 0 vehicles and one Class 5 vehicle were those in which complete measurement of the axle separation was impossible under experimental conditions. Those conditions were either day or night with a two axle tractor pulling a house-bearing trailer (HBT). The wheels of the HBT were difficult to see under optimal conditions in the original experiment because of the extended shadows. An improvement was made to the experimental setup known as test 2, and this test gave better values.

The Class 4 event was a long bus having two rather than three axles.

The Class 5 event not involving the HBT is very similar to that shown at the bottom of page 2. It was evidently called Class 5 because of mistaking the 5 axles on the ground for class number. The two axles recorded were separated by 30.31 units with an overall length of 51.6 units. Each unit is 30.48-cm. The three other events were Class 2 events of limousines.

Of the 1183 ***3 axle vehicles,***
four were outliers.One was a three axle bus that involved no data processing errors. One was an HBT. One was a perfectly processed two axle bus pulling a trailer. The last one was a minivan erroneously identified as a Class 2

3-axle vehicle rather than a Class 3 2-axle vehicle. Three vehicles had been recorded within the same second of time. Two were the same minivan; one had a spurious axle.

From the 779 ***4 axle vehicles,***
there are potential outliers for three Class 5 vehicles, one Class 8 vehicle and one Class 15. The Class 5 vehicles were all designated as pulling a two axle vehicle. They should not be classified as outliers, but rather the last two axle spacings of the pulled vehicle should be ignored as required by the classification scheme. The Class 15 is the vehicle at the bottom of page 2. The Class 8 vehicle mistakenly attributes two axles to a trailing vehicle while only one actually touches ground.

From the 6297 ***5 axle vehicles,***
twenty are potential outliers. Of these, fifteen are of Class 9 one is Class 0, one is Class 11 and three are Class 15. Two of the Class 15 are HBTs without visible wheels. The other Class 15 was a two axle tractor pulling a tridem. The one Class 11 was a three axle tractor pulling a one axle trailer (see below) with a one axle dolly attached to the rear. The Class 0 had no vehicle at the recorded time.

Of the fifteen Class 9 classified vehicles, nine were HBTs; one was a piggyback arrangement similar to the second photo on page 2; one was a four axle vehicle; one was a double trailer; one had a long flatbed tridem trailer; and two were pulling trailers with two separated rear axles.

Of the 111 ***6 axle vehicles,***
eleven were potential outliers. One had 6 axle spacings one of which was negative; one was the Class 15 piggyback shown on page 2. Two were Class 12 piggybacks with a three

axle tractor pulling three 3-axle vehicles; each having but one weight bearing axle. Four were two axle tractors pulling a HBT on a quad wheel arrangement.Two were two-axle tractors pulling two pairs of axles each. One was a four axle tractor pulling a long flatbed with a tandem axle.

Class 15 looked at separately

When it was realized that most Class 15 vehicles were simply unclassified, these were looked at individually. Prior to this discovery a Class 15 was included in the axle investigation providing it stuck out from the other Class 15s. Now all Class 15s are examined.

We had forty-two additional Class 15 vehicles and five Class 10 vehicles to look at, as provided on a list of 45/6 from Ralph Gillmann. Two from Class 15 were duplicates of those already examined, and one had no video tape supplied. The data of 9/9 to 9/11 showed no Class 15 vehicles in the data that included lane changes. It was the no lane change data that had the Class 15s.All but the bottom shot on page 2 and all of page 4 are from the Class 15 list.

Class 10 vehicles mostly the same

Five of six Class 10 vehicles appeared similar to the 9-axle vehicle shown in the first frame of Fig. 1. The sixth Class 10 vehicle occurred on 9/11/93 @ 8:04, but the video tape was not available.

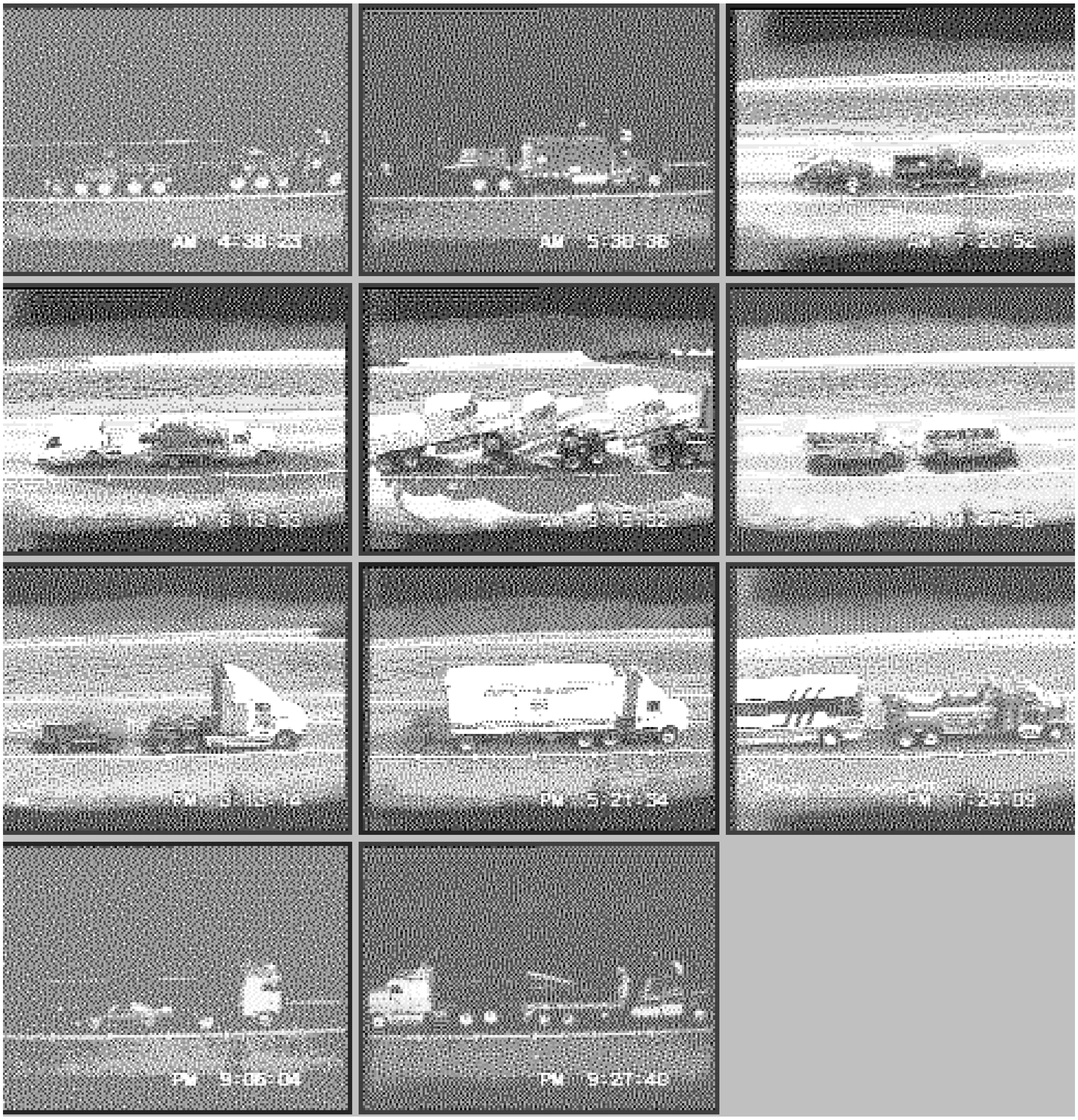


Figure 1. Outlier vehicles

The data and military times of the figures in reading order are:
May 6 @04:38; May 7 @05:30; May 6 @07:20;
May 6 @08:13; May 6 @9:15; May 6 @11:47
May 5 @15:13; May 7 @17:21; May 5 @19:24
May 5 @21:06; May 5 @21:27
During 1996 full sized views were (and may still be) on the web as gif files, e.g.
http://www-xdiv.lanl.gov/XCM/transportation/gifs/1924may5.gif

8/95

FoxPro Database Edited And Sent To FHWA

The FoxPro Database created for the Georgia data has been edited for reclassification of what had been called Class 15 events. These have been reclassified into one of the FHWA 13 classes. The designation of Class 0 remains for those video data vehicle events that have shadows occluding axles. The subclass designation in the Georgia data set will be determined to conform to the conventions used there. For pure FHWA 13 Class classifications, the subclass can be ignored.

Subclass

It now appears that subclass is used differently in Class 8 than it is in the rest of the classes. For classes above 8, the subclass is set to zero. For classes below 7, it indicates the number of axles that are ignored in the classification process. These arise from towed cars or trucks or light-weight hitched trailers. For Class 8, a subclass of zero tends to indicate a three-axle vehicle. A subclass of one or two tends to indicate the number of axles on the Class 8 trailer.

Distinguishing Classes 3&5

An item that needs attention whenever Class 3 and Class 5 data are interpreted, is the classification methodology. These vehicles were still classified when an obscured view of double tires or single tires on the close side of the rear axle occurred. Several conditions served as hints in guessing the classification. First, the presence of an indented hub was taken to indicate a Class 5 vehicle. Next the presence of a flared-out skirt on the back of pickup indicated a dually Class 5 vehicle. A Ford 1 ton truck in the F series is an example of the latter case, whereas a 1/2 and 3/4 ton truck in the same series has but two tires on the rear axle as a standard option.

Class 5 as large

A large vehicle indicated a Class 5 vehicle. Here the term *large* was determined by memory from looking at rental vehicles and other box-shaped vehicles away from the site where a clear view could be had of the number of tires on the rear axle. This approach has a clear basis, e.g. at one rental site in New Mexico a Ryder rental truck that is 8’ tall has two tires on its rear axle, whereas, a ten foot tall Ryder rental truck has four tires. For rental purposes, a variety of loads would be expected. Special cases of trucks designed to carry only foam rubber or styrofoam might not follow such rules.

Are formal rules needed?

We have not checked this version of the methodology with a set of formal rules on the data set. Such rules serve an editing function as a check on the classification by eye. It may not be important to do so. The errors of the vendors, based on this determination of ground truth, are said to be smaller for this distinction than that for distinguishing Class 2 and 3, the car-like versus pickup-like (or van-like) decision.

Nearest neighbor Implications

The nearest neighbor data of Table 1 suggest that about 30 vehicles are in question for four-axle cases when axle spacing data is the only criteria used for classification. These are about 1/3 of all four-axle Class 5 cases and 1/10 of all four-axle Class 3 cases.

Vendor Data Arrives

A total of 11 files of vendor data have arrived at LANL for the time period of 5/5/ -5/7/93. Each file appears to represent the classifications of a single piece of vendor equipment taking data on the same days the ground truth data was acquired.

Most of vendor sets of data are in two temporal parts. A total of 13 files cover the period from 9/9 to 9/11/ 93.

Problems of correspondence

Each event in both the ground truth and/or vendor file is characterized by a time stamp. The number of events in the vendor files do not correspond to the number of events in the ground truth files. Several causes of this discrepancy have been identified after a preliminary examination of one of the larger vendor files.

Top ten reasons

(1) Missing data in the ground truth set when changing video tapes at two-hours intervals.

(2) Lane changing phenomena
whose origin has not yet been fully determined. These include a service truck in one of the data bases and not the other.

(3) Clock differences. The clock times do not uniquely specify an event. There is a time base drift on the order of several seconds per day. A shift of 5 seconds was also observed after a midday 20 minute intermittent off condition of the vendor equipment that skipped 284 vehicles. A similar condition earlier did not result in a clock change. In the earlier case, around 7 AM, the equipment changed to the intermittent state quite frequently. In the mid-day case the equipment appeared to be turned off abruptly, but otherwise functioned much better than the earlier case except for the abrupt time shift.

(4) Long vehicles are sometimes *split* by the vendor and classified as two vehicles. Use of traffic flow rate information might have helped the vendor better classify these vehicles.

(5) Some vehicles registered as vendor vehicle events do not appear in the video ground truth set. They are known as *ghost (vendor) vehicles*.

(6) Some vehicles observed by the vendor do not appear in the ground truth listing, although they do appear on the video. These are called *lost events*. They can be restored to the ground truth set by re-examination of the video data.

(7) Axles appear in the vendor data set that do not appear in the ground truth. These are called *ghost (vendor) axles*.

(8) Dropped axles. These events occur in various forms but seem to occur primarily when vendor equipment is also skipping vehicles.

(9) The number of events in different vendor files vary by 30%.We call the major cause of this condition *dropped vehicle events*, *dropped events* or simply *skipped vehicles*.

(10) Wrong time recorded by hand in ground truth data base. This insidious occurrence gives rise to two vendor errors if not corrected.

9/95

Vendor Data Phenomena

Our primary emphasis in processing the GTRI I20 data set is to ensure the completeness of the ground-truth data set. This function is augmented by studying the vendor files. Some vehicles that appear in the vendor files but not in the ground-truth files belong also in the ground-truth files.

For instance 8 vehicles involved in lane changes were found in the vendors' files without a match in the ground-truth file. One vehicle at 12:58:32 AM on May 6 that did not involve a lane change was missing from the ground truth.

Additionally, the stability of the ground-truth clock can be determined by comparing it to the multiple clocks of the vendors. Establishing the correspondence between the clocks is a necessary step before any conclusions can be reached about the types of errors. Related articles below that discuss this topic further are *Downstream Effects* and *Super Synchronization.* Data regions that have fewer than 7 vendor vehicles for every 10 vehicles in the ground truth set require good if not super synchronization.

In this column, the lane changing effect discussed last month is elaborated upon, and a number of additional effects are described that have been observed in the vendor data files.

Lost lane-change vehicle

The ground truth files are of two varieties: (1) those with vehicles having no lane changes and (2) those with vehicles having lane changes. In comparing many vendors against each other, it makes sense to eliminate a priori all vehicles undergoing lane changes. One vendor's sensors may not experience a lane change, but another's may.

Vehicles making lane changes will eventually be eliminated from the comparison, but such vehicles should appear in the more complete of the two ground-truth sets and not the other. The nature of the error described in last month's Traffic Monitoring Progress associated with lane changes was due to a vehicle that did not appear in the larger lane-change ground-truth set of vehicles.

The vendor's data show that a vehicle should be present in that case for some of the vehicles in the data set. If the vehicle is not in the comprehensive ground-truth set, it is incorrectly classified as a ghost vehicle error, whereas, it is really a lost lane-change vehicle.

Time syntax error

Time syntax errors have been observed in data post-processed by GTRI from the TEL-2CM classifier. It is not clear whether the problem was in the post-processing or the vendor file. Approximately 150 time stamps in the first 48 hour set of data indicate the number of seconds in the hour-minute-second data is 60. An example is 54060, which should be 54100. Corrected, this indicates the fifth hour of the day, the 41st minute and no seconds. A second example in the MIK1 file is 59960 that should read 60000. Correcting these errors then permits a comparison of the vendor data to the ground truth to improve the ground-truth data quality. For instance, a lane change vehicle that was left out of the GTRI ground-truth set was found by these means.

Time carry error: a sequence error

An error that occurs in the GTRI post-processed file GKS2 is caused by an unknown process that recorded serial times out of sequence with the following time stamps on 5/5, 5/5 and 5/7: (141257,141359,141301), (144348,144459,144401),and(112655,112659,112612) respectively. These errors may be found by examining the file for out-of-time-sequence vehicles. They may also be found by examination of those times ending in 59 seconds, but they must be further separated from the many in-sequence vehicles that correctly end in 59 seconds. It is not clear whether the post processing or the vendor equipment produced the errors. It is advisable to re-examine the raw vendor data to clear up this question.

Speeds greater than 100 m.p.h.

A handful of speeds exceeding 100 MPH were found in the DIAS1 file but not the DIAS2 file. These files were recorded by the same machine configured as if two sets of detectors were in two separate lanes. The detectors were actually in the same lane. The wheelbase measurements were not proportional to the estimated speeds recorded in the two files.

This might be explained by the practice of some vendors of recording values appropriate for a passenger car as the default, if the axle spacing cannot be determined.

Downstream Effects

One of the most puzzling aspects of the GTRI I20 data set is the lack of correspondence of the vendor clock reading to the video-ground-truth data-set clock entry.

The clocks in the GKS6000 series detectors agreed well with the video clock after allowing for a slow drift of the clocks with respect to each other. (See the *graphical method* section in the *super synchronization* article for an elaboration of this point.) The clocks of the much further downstream equipment in the 2001 series detectors, however, do not agree well with the video clock after a slow drift time has been accounted for. Furthermore, a correction that is based on the speed recorded in the vendor file and the distance from the vendor equipment to the region recorded by the camera for the GTRI ground truth, a correction improves some parts of the discrepancies in the data, but not all effects are understood.

Full investigation of these effects including examination of the raw vendor data files appears to be more appropriate in a follow-up study. A partial analysis will be given next month. The preliminary mechanisms that could account for the effects are: (1) variations in the speed of vehicles between the ground truth site and the vendor site, (2) post-processing errors between the vendor file and the standardized file, (3) fast time drifts in the vendor equipment or (unlikely) the ground truth clock and (4) unrecorded lane change or weave effects.

Super Synchronization

The two files from the GK6000 equipment, GKS1 and GKS2, were able to calibrate with the ground -truth clock to within an understandable plus or minus one second after allowance for slow time drift. This result is evidence that the ground truth clock has no fast time drift. Closer examination of the data reduces the clock uncertainty to better than one tenth second by the technique we call super synchronization.

Better ground-truth times

The ground truth times were truncated to the nearest second but could be given a precision of one thirtieth of a second or better. The work required to obtain this precision automatically involves additional research; it can, however, be obtained less precisely by hand. It also involves establishing a standard definition of the time for ground truth and vendor alike.

We established by hand, that one can go through the 30 frames between changes in the time-stamp value recorded on the video data. Which of those 30 frames is most appropriate for the time of the vehicle must be established. Five prototype definitions to consider are when: (1) the front of the vehicle crosses a line across the lane at a known location (2) the front axle of the vehicle crosses that line, (3) the center of the vehicle crosses that line, (4) the back axle crosses that line, or (5) the rear of the vehicle crosses that line. Both (4) and (5) suffer from a corollary requirement of clarifying exactly what the rear of the *vehicle* is. Does it include a towed car, for instance, that is ignored in the classification process?

Vendor time conventions also need to be known to make comparisons. Were one vendor to use method (1) and another to use method (5), they would disagree on the time. For a 30 meter (100’) vehicle traveling at 30 meters/second, the two methods would differ by a second. For a passenger car the effect is only one tenth second, but for differences between method (3) and (1) or (5) it is more likely to be 0.05 seconds.

The data reduction used in the GTRI study for ground truth took the starting time to occur when the vehicle was centered in the video frame method (3). If the vehicle was longer than a video frame, method (3) was used only for that part of the vehicle filling the first frame.

Expected errors

The ground-truth video time recorded for the center of the vehicle did not retain the tenths and hundredths of a second. These may be thought of as being truncated before the time was displayed on the frame. The frame time was recorded by the GTRI student in the first 48-hour set of data. In most of the second 48-hour test, the time was obtained by the optical character reader (ORC) weighted mask algorithm developed by GTRI.

When accurate, the information recorded then indicated that the time fell between the value recorded and the next increment a second later. Without reprocessing the video tape, each value within that second interval may be regarded as equally likely. On average, the clock-value recorded will be half a second off. However, if this expected half-second bias is added to the recorded time, the resultant time will differ by plus or minus one half second from the time of the microprocessor clock used to generate the video time stamp.

Rounding considerations

Suppose the time recorded on the video frame is thought of as being rounded to the nearest half second before display, rather than truncated. Then the same plus or minus half second error would result. Therefore, the method of rounding only changes the values recorded on average by the half-second bias. There is no way, by looking at the time stamps printed on the video frames alone, to decide if they were rounded times or truncated times. The two methods simply differ by the half second bias. That is to say, rounded time plus a half-second error in the clock with respect to correct time is equivalent in all respects to time truncated to seconds.

Similar considerations apply to the vendor times recorded. The times recorded in the post-processed standardized file are expected to be plus or minus a half second in precision.

Irrespective of which rounding method is used, the only difference in the error model is an expected bias. Any such bias is accounted for by the same mechanism that allows for the two clocks to slowly drift with respect to each other.

Overall, one would not expect a passenger car to have an error greater than one and one-half seconds. Were this the case, the rounded range difference of unity would suffice to establish correspondence between all of the vehicles. This value was successfully used for the GKS1 and GKS2 files. It did not work for the DIAS1 and DIAS2 files that are for the extreme down-stream data. The explanation for these effects is one of the 4 possible mechanisms described in *Downstream Effects* on page 3 to account for the differences.

Super synchronization

The above estimate of the errors is a worst case scenario. Here we simply point out that there may be extraordinary synchronization between the clocks. In principle, exact (except for fast random errors and unknown and large accelerations) prediction is possible when the time errors are slow. There is no need to allow a two-second difference between the predicted time and that observed.

This type of condition could occur if both the ground truth and vendor round-off procedures were the same and the slow drift between the clocks was a fixed, exact integer. In that case, the clocks would both differ by a constant amount, the integer number of seconds. Furthermore, if one clock used rounding to the nearest half second and the other clock used truncation of all but the integer part, exact correspondence could be expected every integer plus half-second value for the drift between the clocks.

If the drift in time between the clocks were zero, then the error in time that resulted would depend only upon the rounding procedures and the value of the fixed clock difference.

Slow drift prediction

Suppose the slow time drift between two clocks is known and the drift is the only mechanism causing disagreement between the two clocks. Further suppose that the ground truth has time recorded to tenths of a second or better (or infinitely precisely). Then with knowing the rounding off technique of the vendor clock, the vendor time should be able to be predicted exactly within the same precision as the ground truth. Even if the vendor clock rounding off technique is not known, it should not make any difference, providing it is one of the two standard types of rounding off discussed previously. Changing the two assumptions of the rounding off technique simply will change the slow drift bias value by 0.5 seconds.

Graphical method

When the ground truth time, t, is known exactly, a simple graph can be shown to give the delay parameter, d. The difference between the second clock whose time, T, is in seconds and the integer part of the time, t, is plotted as the ordinate and the fraction of a second of t is plotted as the abscissa.

The fraction at which the jump discontinuity of unity occurs in the value of the integer difference of the two clocks is denoted by f. The value of the second parameter M can be expressed as the minimum value of T-[t], the lower of the two steps, where [*n*] indicates the truncated, i.e. integer part of *n*, inside the brackets. M is the height of the lower plateau.

The model is that T=[t+d]. The parameter d is then simply given by d=M+1-f. The explanation of this formula for the differential, d, is best done by example.In any case, points near the discontinuity need to be generated sufficiently well to determine f, the location of the jump discontinuity.

Some properties of the formula can be easily pointed out. First, if the value of d or t increases by an integer, the value of f stays the same, proving that the fractional part of d and t are important in determining f. If t increases by an integer but d is fixed, T increases by that same integer, keeping M fixed and f the same. If d increases by an integer, so does M and the value of d-M is the same as is f.

We show a plot of ten points of the GKS1 data-file data in graphical form in Fig. 2.

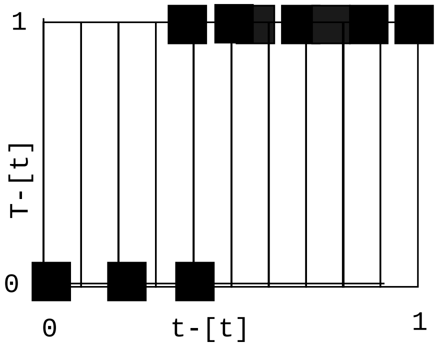


Figure 2. Super synchronization plot

The abscissa values were obtained by plotting the fractional part of the second by counting frames after the change in the seconds on the video frame time stamp; each frame represents an additional thirtieth second, but with overall precision of one tenth second. Note that the precision is limited by slippage in the remote control of the VCR. The frames were advanced until the front of the vehicle reached the white mark near the center of the frame. Here the value of f is localized between 0 and 1 and occurs near 0.4. Choosing an approximate value of f to match the data as 0.4, the formulae gives M=0 and d=0.6.

The actual data are as follows:

Table 1: Raw data for jump plot

GKS1 time	*speed* *MPH*	*GA-time: frame*
144450	58	144449:15
144453	69	144453:1
144459	74	144458:25
144507	69	144507:7
144512	67	144511:17
144530	61	144529:26
144536	63	144535:11
144538	63	144537:29
144552	67	144551:20
144600	69	144600:12

Examination of the plot shows that at an abscissa value near 0.4 there is one point with ordinate of zero and another with ordinate of unity. The one with a value of zero is slightly to the right of the one with a value of unity. This shift occurs because the speed correction has not been applied. The origin of the speed correction is that the GKS1 site is approximately 21 m (70’) downstream from the camera. It takes roughly one second to travel this distance. The vehicle yielding a zero ordinate is travelling 69 miles per hour (MPH). That giving unity is travelling only 63 MPH.The ten percent change in speed translates to a 0.1 second change in time.

This effect moves the faster-vehicle point near x=0.4 and y=0 to near x=0.3 and y=0, thereby rectifying the sequential order. Note that the point near x=0 also would be moved 0.1 to the left, but this would not affect the prediction of the vendor time because it did not move past the jump discontinuity.

With d=0.6, all times having their fraction of a second between 0 and f=0.4, would when added to d, give a value of T equal to the integer (seconds only) part of t plus the integer part of d. Those times falling in the last part of the second beyond 0.4 seconds would produce a value of T equal to the integer part of t plus the integer part of d plus unity.

Another way to visualize the clock synchronization is as two scored rulers that have slid with respect to each other. The amount of the slide corresponds to the clock differential d.

This super synchronization method may be easily automated by programming for a slow drift of the discontinuity location, f. This technique should become a standard practice and is even more useful with vendor times to a fraction of a second that in principal are available before they are recorded in seconds. With many vendors recording times, the super synchronized time should be able to be propagated down stream from one vendor site to the next.

Split-Vehicle Lost Axle Count

One kind of equipment failure was found in equipment using a loop-piezo-loop configuration. This condition was a vehicle split but with a lost axle count on the second vehicle.

A split vehicle is a vehicle that is recorded as two vehicles although it is only one vehicle.

The phenomena could be accounted for by the classification device being able to detect the front two axles but not the rear two axles. In a loop-piezo-loop configuration the piezo detector is the only part of the device that can measure the time between tires passing over the line of the piezo detector transverse to the highway.

An example of 5/5/93 data shows two vehicles at the same time of 170606 and same velocity of 62 miles an hour. The first vehicle had a length of 14’ and the second 11’. The wheelbase was 8.5 for the first vehicle and 0 for the second which was listed with 0 axles.

Jerry Schiff’s Observation

As a sales person, the value of a study of “outmoded equipment” may appear to be of dubious value. Jerry Schiff, a sales person associated with the GTRI I20 study, has estimated that “Only 50-70% of the [type of] equipment [covered in the Georgia study] is in service today.” Ralph Gillmann, of FHWA replied that the I20 study has served as an impetus to develop new equipment: “A vendor that came out poorly on the Georgia test would [want] to make some changes and announce a new, improved product.”

The results of the GTRI study certainly do not indicate all the improved performance characteristics of automatic traffic recording equipment that can be purchased today. The memory limitations of processors used in those days, for instance, have been completely overcome.

10/95

Performance Matrix Evaluated For Scheme F

Scheme F has been adapted by many vendors as a standard method for classifying vehicles. Sometimes the vendors will modify but one or two of the criteria. We present a comparison of the Scheme F performance with respect to the ground truth that occurred in the Georgia study.

Edited data from the GTRI data set of 1993; traffic classification are shown as Table 2 and Table 3. The Scheme F implementation that we used is shown on page 6. The May data shows strong asymmetries in the Class 2 and Class 3 matrix elements. Of the 8565 vehicles identified as Class 2 in the ground truth data set, 243 (2.8%) were classified as Class 3. However, of the 5560 Class 3 vehicles 3018 (54%) were identified as Class 2. The corresponding percentages are 2.3% and 58% in the September data.

Table 2: M_{ij} for Scheme F for the May 5-7, 1993 Georgia test

Truth\SchF	1	2	3	4	5	6	7	8	9	10	11	12	13
1	18	2				1							
2		8316	243			1		5					
3		3018	2443		4	3		87			5		
4			3	60	6			1					
5		22	361	93	235	1	9	22			16		
6			3	20		461		1	14	1		1	
7													
8			3					319					
9									3671		5		
10										31			12
11								3	1		101		
12												29	
13													1

Table 3: M_{ij} for Scheme F for the September 9-11, 1993 Georgia test

Truth\SchF	1	2	3	4	5	6	7	8	9	10	11	12	13
1	30	3											
2		10128	234			2		9					
3		3700	2495		44	3	1	79			3		
4			2	44	5	4							
5		7	145	62	149		6	18		1	7		
6			2	10	1	191		3	5				
7							1	2					
8		2	6			1	1	278					
9									3152	1	11	1	
10										27			9
11											93		
12												28	
13													

The problem with the asymmetries was reported previously. There simply is no cancellation of errors. That is, in a large sample of vehicles such as the May data, the actual number of class 2 vehicles is approximately 8316+243, whereas the measured number is nearly 8316+3018. Had the off-diagonal matrix elements been symmetric, these errors would have cancelled.

Note that the definition of Scheme F is given in report 84-5 entitled *Field Evaluation of FHWA Vehicle Classification Categories*, of the Materials and Research Division, Maine Department of Transportation, January 1985. That report leaves several minor ambiguities in the definition of its classes having to do with whether or not end points of axle length ranges are included as part of the class definition.However, some of the end points are well defined, and in one unfortunate 4 axle case they cause the extension of a class in a very awkward fashion.

When the first axle separation is between 10’ and 15’ and the second separation is greater than 5’, the 4 axle vehicles having the last axle separation of exactly 3.5’ slip through one condition requiring the separation to be less than 3.5’ (these would end up as class 3) and another condition for the separation to be greater than 3.5’ which would be classified as class 8. They become classified as class 7. Plotting the classes on a two dimensional grid with the second and third separations serving as the x an y axes, most of the regions are in rectangular blocks, but class 7 extends along the class 3 and class 8 boundary along an isolated line.

Super Synchronization

Last month we reported the phenomenon of super synchronization and indicated that we would investigate its applicability to a set of GTRI acquired vendor data. Super synchronization is applicable to comparison of traffic classification by multiple vendors each with their own clocked sensors situated at different locations downstream from the master clock location.

In the case of the Georgia I20 study, a preliminary look at the super synchronization data suggested that there were inconsistencies that needed to be addressed. Here we remove the inconsistencies associated with vehicle length by studying passenger cars only. A passenger car has a length that requires about one-tenth second for it to pass a given location. During this time, three frames of a thirty-frame per second video camera elapse. Furthermore on our current equipment there is about a three frame experimental error in determining the frame number of an event with respect to the origin frame. The *origin frame* is the first frame of a sequence of thirty frames all having the same integer number of seconds recorded thereupon.

The idea of super synchronization is quite simple. The master clock has a time that is determined by the integer number of seconds recorded on a video frame and by the number of the event frame with respect to the origin frame. The vendor clock is modeled to have a slow drift that involves a calibration change of something like a second per hour. Otherwise the vendor clock measures an event related to the master clock event but at a later time.

The time interval between events is the time it takes the vehicle to travel from the master clock to the vendor site, L/S where L is the distance downstream and S is the vehicle speed in compatible units, assumed constant for each vehicle in this study. Also, the slow drift is present of d(t). Thus, we have $T_v = [T_m + L/S - L/S_0 + d(T_m)]$. Where the brackets indicate the integer part of the quantity. The term L/S_0 is simply for convenience and may formally be regarded as part of $d(T_m)$. This quotient, L/S_0, is a reference time required to travel to the vendor site based upon a reference speed S_0 which we take to be 31.3m/s (70MPH). The equation for T_v simply states that the vendor event time is the sum of the master clock time plus the relative clock drift plus the transit time offset to zero at a speed of $S_{0.}$

All the events we study here are from the data of 9/10/93 in the time interval [144450,144600]. The reason this region was chosen is that most vendors had 12 vehicles recorded. Of these twelve, seven turned out to be passenger cars. Even in this relatively good interval, one of the vendors had incomplete data for passenger cars.Another vendor did not report speeds at all. These two vendors were not included in Table 4 and Table 5 below. The raw data is shown in Table 4. In Table 4, L indicates the length downstream and F indicates the frame number past the origin frame, where the origin frame was counted as zero. The seven F values are taken from the video and are used to compute the fraction of a second for the master clock time. The six L values indicate that six vendors were used in this study where the lengths, L, are measured in 30 cm units (1 foot).

The upper of the two number pairs recorded in the matrix indicates the difference in the vendor recorded time (post-processed to be in integer seconds, see Sept. Traffic Monitoring News) and the integer seconds time recorded on the video frame of the event of the front of the vehicle crossing a fiducial line.

The lower of the two numbers is the post-processed vendor recorded speed of the vehicle measured in units of 0.447 m/s (1 MPH). One exception is the speed recorded in the shaded cell of Table 4. It was originally listed as 29 and was judged to be in error and was replaced by its neighboring value of 70. Speeds were rounded to an integer value.

Table 4: Raw integer times and speeds for extended super synchronization study. L is distance downstream, F is frame number. Seven vehicles at six locations.

L\F	*0*	*2*	*19*	*20*	*23*	*24*	*26*
0	5 73	6 69	6 73	6 72	6 67	6 68	6 73
70	-1 72	-1 69	-1 71	-1 72	0 68	0 68	0 70
200	0 69	0 67	0 69	0 69	0 66	1 65	0 67
450	-1 71	-1 68	-1 71	-1 70	0 68	0 66	0 70
550	0 71	0 67	0 70	0 70	1 69	1 67	0 61
650	-1 71	-1 69	-1 71	-1 71	0 69	0 68	0 69

Table 5 shows that super synchronization holds for these vehicles except for the shaded regions. At

Table 5: Effective Frames vs Integer Time for seven cars at six locations

car #	*down-stream distance*	*effective frame*	*integer time*
0	0	0.0,	5
1	0	2.0,	6
2	0	19.0,	6
3	0	20.0,	6
4	0	23.0,	6
5	0	24.0,	6
6	0	26.0,	6

Table 5: Effective Frames vs Integer Time for seven cars at six locations

car #	*down-stream distance*	*effective frame*	*integer time*
0	70	-0.6,	-1
1	70	2.3,	-1
2	70	18.7,	-1
3	70	19.4,	-1
4	70	23.6,	0
5	70	24.6,	0
6	70	26.0,	0
0	200	0.8,	0
1	200	4.6,	0
2	200	19.8,	0
3	200	20.8,	0
4	200	26.5,	0
5	200	28.5,	1
6	200	28.6,	0
0	450	-1.9,	-1
1	450	5.9,	-1
2	450	17.1,	-1
3	450	20.0,	-1
4	450	26.9,	0
5	450	32.0,	0
6	450	26.0,	0
0	550	-2.3,	0
1	550	9.2,	0
2	550	19.0,	0
3	550	20.0,	0
4	550	25.3,	1
5	550	31.2,	1
6	550	49.7,	0
0	650	-2.7,	-1
1	650	4.8,	-1
2	650	16.3,	-1
3	650	17.3,	-1
4	650	25.8,	0
5	650	29.6,	0
6	650	28.8,	0

each fixed downstream distance, there are results reported for seven vehicles. The integer time is taken from Table 4. The effective frame represents the frame number that would have been observed by a camera had it been located at the vendor site. The effective frame is the same as the master clock frame if the downstream distance is 0 or the vehicle speed measured at the vendor location is S_0. The transit time is converted into lapsed frames by multiplying by 30 frames per second (fps). This time is then added to the master clock frame after subtracting off the time it would have taken a vehicle traveling the distance L at speed S_0.

Compliance with super synchronization is defined to occur if the integer times are in non-decreasing order after sorting the effective frame values in ascending order. This condition applies for vehicles at a fixed downstream location. It is the same as requiring a plateau of the variety reported last month. Shaded entries in Table 5 that do not comply with this criterion are with (car number, downstream distance) pairs of (5,200),(6,200) and (6,550). The first two pairs would be in order were they switched. A frame-measuring error of only a fraction of a frame accounts for this discrepancy. The (6,550) out of sequence entry apparently is bad data. Ascribing this category to the data has additional supporting evidence. The vendor reports the vehicle to be a 5 axle (non-passenger car) vehicle and gives 0 for all the axle separation distances. This is in contrast to the other vendors who report the vehicle to be a passenger car.

This last condition, which is a new type of vendor fault, also suggests a possible root cause operational phenomenon of the vendor processing. Exactly what value does the vendor report for a vehicle time? Does it include processing time used in determining the separations? Certainly for data restricted to passenger cars only, we have shown that super synchronization holds within experimental errors for the data set selected. Any vendor processing time for passenger cars appears to be negligible.

We have shown that a set of vehicle data is consistent with super synchronization for various downstream sites. The intrinsic time error with super synchronization is a fraction of a second. Exactly what fraction depends on the details of the vehicle transit history. This error is certainly less than the 3 second error assumed in the GRTI analysis. Matching up data from ground truth site to vendor site in which the fraction of vehicles detected by a vendor is less than 70% requires a high-precision technique such as super synchronization.

12/95

Reinventing The Parser

Early work done on the parser that takes sensor events and converts them to vehicle events has been giving sporadic results especially for large numbers of axles. Here we take the philosophy of reinventing the parser so that we can explain its features in a motivated way. We use some data known as run R257, lane 1.

In the course of this explanation, it has become apparent that a major additional software tool is needed to visualize the parsing process. Not only do the anomalies of one lane of Run R257 need to be understood, as we explain below, but many runs of data need to be examined. To examine the data for a single event, both in the sensor realm and that of the video ground truth requires about 30 minutes. Putting the data together by a computer application has the potential of cutting this time by a factor of a thousand. This type of gain requires a good means of visualizing the entire process and is beyond the scope of this December report. We do, however, report some of the preliminary results obtained in December.

Two piezos and two loops

We will start with a system that has two piezo and two loop sensors in a PLPL arrangement where P indicates a piezo sensor and L indicates a loop sensor. The study of two piezos and two loops also includes the other cases with fewer sensors because of the possibility of lost axle or loop data.

The L center location and the P location are given in sequence as downstream distances in meters from a fiducial line transverse to the lane. The downstream distances are: 0.0, 1.71, 3.38, 6.58. The loops begin 1.2192-m before center and extend 1.2192-m after center.

Now, let us examine some sequences of sensor events that are within one second of each other. Here for the loops we use the first signal sensed as the start time. The second break between events divides the sensor stream into groups of events we call time blocks. These blocks are shown in Table 6.

Table 6: Sensor Blocks above thresholds

Pth/Plo	*Lth/Llo*	*Sequence*	*Block number*
3/68	0/59	P1L1P1P2P2L2	1
6	0	-	2
2/125	0/18	P1L1P1P2P2L2	3
11/50	0/56	P1L1P1P2P2L2+	4
11/202	0/30	P1L1P1P2P2L2	4b
7	0	-	5
10/136	0/54	P1L1P1P2P2L2	6
6/44	0/59	P1L1P1P2P2L2	
0/135	0/59	P1L1P1P2P2L2	7
9	0	-	8
12	0	-	9
12/133	0/45	P1L1P1P2P2L2	10
10	5	-	11

Table 6: Sensor Blocks above thresholds

Pth/Plo	*Lth/Llo*	*Sequence*	*Block number*
3	0	P1L1P1P2P2L2	12
3	0	-	13
3	0		14
8	0	-	15
0/96	0/28	P1L1P1P2P2L2	16
10/96	0/50	P1L1P1P2P2L2+	17
10/138	0/26	P1L1P1P2P2L2	17
6/72	0/45	P1L1P1P2P2L2	b
15	0	-	18
4/103	0/65	P1L1P1P2P2L2	19
2	0	-	20
4	0	-	21
7	0	-	22
4	0	-	23
6	0	-	24
14	0	-	25
9/8	0	P1L1P1P2P2L2P1P2*	26
6	0		27
5	0		28
10	0		29
3/81	0/11	P1L1P1P2P2L2	30
8	0		31
7	0		32
13/134	0/37	P1L1P1P2P2L2	33
5/119	0/22	P1L1P1P2L2P2*	
9	0		34
14	0		35

The first time block is P1P1L1P1P2P2P2L2P1P1P1. If we remove all P events with amplitudes of 3 or lower, where the maximum in the first block is 122, the sensor stream is P1L1P1P2P2L2, the sequence listed in the table. This stream has two P1, two P2, one L1 and L2 sensors. The pattern is typical for a vehicle whose axle spacing is less than the downstream spacing of the piezo sensors. Here a + indicates that the block is split in two when only above threshold events are used. A value of 0 for threshold in either Pth or Lth indicates that there are no noise pulses of that variety P or L in the retained signals respectively. Following the slash is the lowest value of the sensor signal above threshold of the signals retained.

When two lines are used for the sequence, the sequence is not broken but is continued as shown. Then the low value appears on the line associated with the block in which it occurs. A dash for the sequence indicates that all the entries in the sequence are eliminated when the above threshold condition applies. In that case, there is no Plo or Llo value to report.

Table 6 shows the first 35 blocks. It is seen that 22 out of the 35 blocks are pure noise. These noise pluses are easily removed. Most of the blocks are two-axle vehicles. However, some blocks show special patterns shown in Table 7 that involve above-threshold sensor signals that are not simple two-axle vehicles. These four blocks are numbered 6, 17b, 26, and 33. There are 11 two-axled vehicles that make up the rest of the cases shown in Table 5.

Table 7: Special Entries

Pth/Plo	*Lth/Llo*	*Sequence*	*Block number*
10/136	0/54	P1L1P1P2P2L2	6
6/44	0/59	P1L1P1P2P2L2	
10/138	0/26	P1L1P1P2L2P2	17
6/72	0/45	P1L1P1P2P2L2	b
9/8	0	P1L1P1P2P2L2P1P2*	26
13/134	0/37	P1L1P1P2P2L2	33
5/119	0/22	P1L1P2P1L2P2*	

Block 6

For block 6, it takes about 0.08-s for the second axle to follow the first and about 0.09-s for the third to follow the second. But it takes 0.82-s between the second and third axles. The L1 signal is 0.11-s long and the L2 signal is the same for the first unit. The second unit has L1 0.14-s long and L2 0.13-s long. All of this suggests that we are probably dealing with a pair of two-axle vehicles. Otherwise, these would have to be a unit with a very long hitch that does not register on the loop detectors. This opinion is verified by the ground truth.

Block 17b

For block 17b the first axle spacing is 0.11-s; the second is 1.06-s and the third, 0.07s. The loops for the first unit average 0.155-s and for the second unit 0.09-s. This is even more clearly two separate units than in the block 6 case.

Block 26

The block 26 case is very interesting and the detected signals are indicated in Table 8.

Table 8: Block 26 data

sensor	*amplitude*	*Lti*	*center time*
P1	4		0.15
P1	11		0.60
L1	44	0.13	0.74
P2	3		0.70
P1	9		0.70
P2	91		0.74
P2	87		0.84
P1	8		0.85
L2	45	0.13	0.93
P1	9		1.41
P2	9		2.09

The shaded cells are the ones utilized and the unshaded cells are noise cells. The noise level is at 9 and the lowest signal level is also at a value of 9. This event also is discussed in the article *Parser Tool Aids Analysis*.

Note the consistency of the event. First it fits the canonical pattern of P1L2P1P2P2L2. Second the verification of the equality of the time intervals between the two P1 sensors and the two P2 sensors are equal at 0.10. This number is consistent with the length of the loop signal at 0.135-s. The time between the first P1 and the first P2 sensor is 0.134-s. The time between the first L1 and the first L2 event is 0.19-s, approximately 42% longer. The piezo spacing is 3.38-m and that for the loop is 4.87-m or 44% longer.

This vehicle is one in which one of the piezo signals that is essential for the identification of the vehicle falls below or at the noise level. This constitutes a new fault mode described below.

Block 33

The block 33 signal shown in Table 9 has an unusual first unit pattern with the last two L2,P2 events reversed from the usual pattern. This may indicate a long overhang trailing the second axle. The differences between the leading half-height of the unit one P2L2 is 0.02-s and that of L2P2 in the second unit is 0.001-s. The time used to order the events for the loops is roughly (center time) -(1/2)Lti.

Table 9: Block 33 data

sensor	*amplitude*	*Lti*	*center time*
P1	2		2.91
P1	2		3.66
P2	3		4.04
P1	9		4.27
P1	2		4.97
P1	13		5.31
P1	134		5.61
L1	38	0.14	5.74
P1	144		5.71
P2	275		5.73
P2	311		5.84
L2	37	0.14	5.92
P1	5		6.21
P1	148		6.42
L1	22	0.17	6.58
P2	119		6.55
P1	196		6.56
L2	20	0.17	6.77
P2	241		6.69

The first axle spacing is 0.10-s; the second 0.71-s; the third 0.14-s. The loop durations are consistent with those at 0.14-s and 0.17-s. Thus this is similar to block 6 and block 17b. All vehicles examined so far are from the total of 18 two-axle vehicles in this study.

New piezo fault mode

Of all the above vehicles, only one suffered from a low minimum piezo signal to a maximum piezo noise ratio. the S/N ratios were at worse 50/11 =4.5 in block 2 except for block 26 where the ratio was unity.

In block 26, not only was one signal low, another signal was but 20% above noise, too. These both occurred on piezo P1. It thus appears that P1 misfunctioned during that time. Examination of proceeding and following vehicles show that the signal was fully operational within seconds. It is doubtful that both axles were unloaded by a quirk double bounce as they passed P1. It also seems unlikely that the explanation could be a change in the traction of the front wheel by accelerator pedal changes. Such changes would be expected to occur gradually over several tenths of a second. The fact that the vehicle was a compact passenger car, may suggest that a mechanical loading threshold may be involved. Variations of sensitivity of a piezo detector with respect to transverse lane position may also play a part.

Note that similar intermittent phenomena have been observed for piezo detectors where the inoperable time is considerably longer. Mitchell Latta associated with the Idaho Department of Transportation reports that Steve Carlson has revived strip piezo sensors that have been inoperable by the application of a 9 Volt battery across the piezo’s leads. Mitchell’s hypothesis is that there is some kind of electro-mechanical crystalline phenomena taking place.

The purely parser question is whether the sensor events at 579-s by the sensor clock in run R257, lane 1 correspond to a single vehicle or multiple vehicles. Such a question is in principle describable by a forward model prediction. It may not have a unique answer, but may have to be answered in a probabilistic sense.

Parser Tool Aids Analysis

The analysis of sensor events into parsed vehicle events can be done with a model of the data system. Observation of actual events suggest a wide range of phenomena occurs for the events. These events are often difficult to comprehend without the observations which at times have been unpredictable. That is to say, the observation drive the model, or the model is phenomenological.

Construction of a tool to visualize the parsing process has enabled examination of all the data in run R257 taken with our equipment in Pojoaque NM.

The phenomenological model is primarily concerned with the noise and dropped or weak signals. Under what conditions do they occur? Are there any predictable features of them?

What are some of the run R257 features? There is: an isolated low amplitude piezo signal at the time of 570.0-s that barely is above the threshold requirement — but is real. There are real loop signals that are dropped at 441.9-s; then at 152.13-s there are two piezo noise pulses; and at 452.7-s there are four above threshold pulses that are actually noise.

The shading of the cells in Table10 indicate that the two vehicle sensor patterns are actually two separate vehicles as determined by the ground truth shown in Table 11 where the same shading occurs in column one only.

The thick boxed cells in Table 10 indicate piezo signal problems. There are two boxes with dropped signals and two with added signals. The lightly outlined boxes in Table 10 occur where the lane position is not normal, as specified in Table 11.

The double outlined boxes indicate a single vehicle. At 663-s, there is a passenger car with a long trailer referred to as Class 2E. Except for the E that indicates a trailer, A,B,C, and D indicate increasing sizes of vehicles within the FHWA class. For the data only, Class 2 and Class 3 vehicle events occur.

In the ground truth table, Table 11, an asterisk indicates that the vehicle was in an abnormal lane position. This condition accounts for four of the seven events with lost signals. At STS12 where the data were taken, the lane changes are higher than normal.

Some of the numbers derived from the piezo signals and the loop signals are redundant. They all can be used to obtain velocities. Two values of the velocity based on the piezo signal are possible, and two from the rise and fall of the loop signals. The four values can be compared. These comparisons then give the three numbers: the piezo contrast ratio, the loop contrast ratio, and the piezo-loop consistency ratio. All three of these numbers needs to be small for a good parsing of the vehicle into an axle pair.

The combinations that can indicate a good two-axle group are: 2P2P1L1L, not necessarily in that order. Speeds can still be obtained for one 2P event to be replaced by a 1P event, or for one 1L event to vanish. One can also tolerate the loss of the first P1 signal and the second P2 signal; this is called a crosswise piezo loss. Such a crosswise piezo loss does not permit a piezo-based computation of the vehicle speed.

The number of redundancies in obtaining a sensing of two axles is 3 for all sensors. It is 2 for the loss of a single piezo signal. It is 1 for the loss of one loop or the loss of a signal from both axles of the vehicle in any piezo combination.

No redundancies occur when both a loop and a single axle on one piezo is lost. For losses more extensive that these, the computation requires more information than what we plot as the primary information.

Although the loss of a single loop leaves only one redundancy for parsing axle pairs, loops are useful when the process involves more axles than two. A loop gives a measure of the overall length of the vehicle. It can be useful in deciding if a tractor plus tandem semi is really two vehicles or one. Two separate vehicles would have considerably shorter magnetic lengths.

Examination of vast quantities of data are necessary for having an understanding of the dropped signal and noise pulses.

Table 10: Special Events in R257 (far lane) Thick/thin box for piezo/lane problem. Double box for vehicle with trailer. Light shading for a pair of vehicles.

time front begin	*pattern front*	*time back begin*	*pattern back*
19.33	4P2L	20.24	4P2L
82.67	2P2L		
105.61	4P2L	106.4	4P2L
151.41	4P2L	152.13	2P
161.63	4P2L	162.36	4P2L
263.49	4P1L		
373.87	4P2L	374.46	4P2L
396.89	2L	397.5	4P2L
441.93	4P		
449.39	4P2L	449.77	4P2L
452.7	4P		
505.09	3P2L		
570.	1P		
579.32	4P2L	579.55	4P2L
663.72	4P2L	664.07	1P1L
846.36	4P2L	847.34	4P2L

Table 11: Ground Truth from Video (Lane Not Normal Indicated by *)

front time	*Classification*	*Video time #1*	*Video time #2*
19.33	2B,2C	17.1	17.8
82.67	2B	80.1	
105.61	2D,3C	103.2	104.3
151.41	2C		
161.63	2C,2C	159.2	160.0
263.49	2C*	261.2	
373.87	2D,2C	371.5	372.0
396.89	3D*,3B	394.3	395.3
441.93	3C*	439.5	
449.39	2B,3B	446.8	447.8
452.7	none		
505.09	3A	502.6	
570.	3B*	567.2	
579.32	3B,2B	576.8	577.1
663.72	2E	664.8	(+3.5)
846.36	2B,3C	847.3	848.4

5/96

Georgia Data Analyzed

Preliminary analysis of the Georgia I20 data set was performed with FOXPro. These results were reported last fall.One of the shortcomings of that analysis is that 4-6 hours were required to complete the FoxPro analysis. The running time requirement made it slow and difficult to make modifications and improvements in the algorithm. Furthermore, this amount of time was considered to be unreasonably long compared to the neural network classification time (the same data set took on the order of 5 seconds) reported at NATDAC. The long running time made it inconvenient and impractical to test new approaches to processing the data sets.

The FoxPro program was translated into C++ by invoking a newly designed class of objects. These objects read in a page of data at a time and provide random access to the data on the page. The page object also has functions to look ahead and reposition the current record attention to a new data record anywhere down stream. Functions were added to make the interface be easy to translate from FoxPro. The running time was 50 seconds on a Sparc 10 and a little under two minutes on a Sun IPX.

Improvements to the algorithm were made that accounted for the phenomena of increased transit time of slow vehicles.The criterion for a split vehicle was altered in two ways: (1) to require the axle count of candidate split vehicles to add up to the ground truth count, and (2) to allow up to a time difference of a second in the recorded vendor time of the two vehicles being considered as a possible split. The later condition was added because vendor times were quantized in seconds and not all splits occurred with the same integral number of seconds.

Three vendor data sets were examined. The vendor located at 300 (30 cm/unit) units or 90-m downstream was examined first. Except for 7 events, all were handled by the program.There were approximately 500 exceptions for the set located 650 (30 cm) units downstream. And 800 exceptions for the vendor located 850 (30 cm) units downstream. The exceptions represent three to six percent of the vehicles.

The exceptions are primarily lane changing events that have not been included in the main file for the ground truth data set. The video was examined for the first dozen of the 500 exceptions. Two types of exceptions occurred. (1) insufficient time error criterion and (2) missing lane changing events in the ground truth database. Two events fell in category (1) and eight in category (2). However, the lane changing video was not on hand to verify the condition. What was established is that vehicles were in positions in the far lane to switch into the near lane and account for the extra vehicle.

The data examined from the 800 exception set could not verify the diagnosis because there was an intermittent problem in the vendor equipment. The equipment had fairly good temporal registration of events with the ground truth, but for two extensive periods within the first set of data, the axle separations and axle counts appeared completely unrecognizable when compared with the ground truth. A comment was made in the Georgia study report: “On a later visit, the vendor replaced the EPROM with updated version and readjusted the sensor threshold.” This alteration could account for the improvement seen before the second problem period arose.

A time period was chosen during which both vendors were obtaining good axle data, i.e. data in substantial agreement with the ground truth. If vehicles in the far lane ground truth video appeared in the near lane 650 units downstream, they should also appear at 850 units downstream.

The first possible ten events were chosen for the comparison. Of these seven appeared in the second data set. The times in hours + minutes + seconds are: 171833, 171845, 172812, 173820, 174032, 174044, 174301, 174657. One of the three other events was associated with a tape change. One was associated with overly stringent time agreement criteria. In one case the problem was attributed to the aftermath of a previous error. A five axle vehicle had mistakenly been said to have ten axles. The time event after that occurrence was delayed abnormally by the vendor equipment for about two seconds, thereby destroying temporal registration.

Video verification was undertaken to see if the seven events were possible under an assumption of vehicles traveling at constant speed. In most cases the vehicle in the far lane had already passed. It simply had to change lane where no interfering vehicles were present. In all but one of the remaining cases, the vehicle could pass the slower vehicle with at least 30 units of headway, before arriving at the vendor station.

For one event, the assumption of constant speed was shown not to hold. The speed at the vendor station was recorded as 63 MPH, but was estimated to be 8% higher based upon measurements from the video at the location of the ground truth camera. Were the 8% higher figure to hold throughout, the passenger car still would not quit have sufficient headway to pass the three axle slower vehicle. Additional acceleration has to be hypothesized for this event for it to make sense. The event is recorded as being in the near lane at both vendor locations with approximately the same speed.

No ghost vehicles observed

Contrary to results reported in the GTRI study, we do not see significant cases of the vendors reporting extra vehicles. In our study, these events would show up as extra-vehicle exceptions from the ground truth data base. As reported above, the explanation of these extra-vehicle exceptions is primarily from unrecorded vehicles undergoing lane changing subsequent to the ground truth observation by the primary camera, an effect not described in the GTRI study.

Furthermore, for the vendor located 90-m downstream where lane changing is not expected to be an important effect, we observe only seven exceptions that could possibly be extra vendor vehicles. None of these fall into the extra vendor vehicle category. Two are slow utility vehicles. Two are lane-changing vehicles not in the database. The remaining three are vehicles that the vendor split but that no longer meet our split vehicle axle count criterion. The vehicles in question were long trailers carrying houses. Because of the lighting, it was impossible to count the number of axles for the ground truth. Although the vendors split these vehicles as two separate vehicles, the axle count didn’t agree as we require for an official split vehicle.

The GTRI study reports 1339 extra vendor vehicles contrasted with the no extra-vendor vehicles that we find for this 90-m downstream location in the first 48-hour test.

7/96

Performance Of The Calibrated Georgia Data Set

Two different subsets of data have been used in evaluating calibrating techniques for the Georgia data set. These calibration techniques are for two-axle binned vehicle data. The remaining data are used for testing how well the calibration actually performs. The calculations were carried out on an AIX PowerStation 590 computer.

Method I chooses a training set at random from the entire database. Likewise the two testing sets are chosen at random. Method II chooses the May 5 1993 (first day) data for calibration. The testing sets are for each of the remaining days, May 6, May 7, September 9, September 10, and September 11.

In this study, Classes 7 and 13 have too little data to take the numbers seriously. In all cases of other classes being misidentified as Class 7 or Class 7 as other classes, the E1572 algorithm can be modified to perform perfectly on the Georgia data set in three ways: (1) A6 replaced by A2*4 whenever (see Fig. 3) they occur and (2) A3 be checked for axle spacing of the tridem to make sure it is under 5’, and otherwise classify it as A1*11 (3) correct the ground truth database for vehicles classified as Class 7 that should be in another class such as Class 8 or remove those in the ground truth database that do not exist on the video. Another caveat about Class 7 involves the vehicle shown in Fig. 4, it is classified as B2 and thereby Class 7. It is not clear to me whether this is a single unit vehicle or not. If not it would be an A1*2 and another rule could be required and another correction could be made to the database.

The tabulated results

In Table 12, the row number of the data is the FHWA class, thus, 6260 is the number of Class 2 vehicles in the ground truth for the training set. Testing with the training data yields few errors.Tables 13 and 14 are for the two random mutually exclusive samples of roughly 1/3 of the data each:

Table 12: Errors with Testing = training = geo.trn, Classes 1-13

no. in ground truth	*%hybrid*	*%Scheme F*
15	0	-7
6260	0	32
4005	0	-49
42	0	128
390	0	-60
242	0	-5
1	0	700
220	0	36
2212	0	-1
30	0	-40
70	0	21
21	4	9
1	0	1300

Table 13: Training = geo.trn, testing=geo.tst, Classes 1-13

no. in ground truth	*%Hybrid*	*%Scheme F*
21	-10	-20
6417	-1	32
3908	0	-52
34	9	161
378	-2	-61
246	-6	-9
1	-100	200
192	4	34
2331	0	0
27	-1	-4
68	1	23
20	0	0
0	/0	/0

Table 14: (geo.trn,geo.test)

no. in ground truth	*%Hybrid*	*%Scheme F*
18	-3	-6
6261	0	33
3972	-1	-51
49	-20	112
386	-6	-64
224	2	-7
1	100	600
198	2	38
2295	0	0
22	-15	-14
60	3	21
16	1	0
0	/0	/0

The first conclusion is that with representative training data on representative test samples, the Hybrid method has at most (excluding Class 7 because of previous remarks) 15 to 20% errors whereas Scheme F has greater than 100% errors.

Tables 15-16 uses representative samples for training but applies it to the two successive days. Then we predict the results from September using the same training data geo.trn data in Tables 17-19. There follows similar results using training with data from 5/5 only in Tables 20-24. Similar results are shown in Appendix B for Tables 25-26 for training with 9/9 data.

Training with random data

Table 15: (geo.trn,geo.56) for 5/6/93

no. in ground truth	*%Hybrid*	*%Scheme F*
8	6	0
4203	-2	31
2762	4	-45
26	12	200
356	-14	-64
208	-2	-5
0	/0	/0
164	7	38
1860	0	0
29	2	-21
58	-5	8
15	1	6
0	/0	/0

Table 16: (geo.trn,geo.57) for 5/7/93

no. in ground truth	*%Hybrid*	*%Scheme F*
5	-20	-20
2412	1	35
1676	3	-43
33	-17	96
278	-32	-75
202	-1	-5
0	/0	/0
84	2	38
1026	0	0
9	5	-34
30	2	40
10	-5	0
0	/0	/0

Table 17: (geo.trn,geo.99) for 9/9/93.

no. in ground truth	*%Hybrid*	*%Scheme F*
5	-20	-20
2126	-2	29
1214	0	-51
6	56	266
95	30	-45
48	6	0
1	0	200
88	-4	18

Table 17: (geo.trn,geo.99) for 9/9/93.

no. in ground truth	*%Hybrid*	*%Scheme F*
782	-1	-1
6	20	-17
16	3	12
6	-5	0
0	/0	/0

Table 18: (geo.trn,geo.910) for 9/10/93.

no. in ground truth	*%Hybrid*	*%Scheme F*
17	2	-6
5081	1	33
3172	-6	-55
27	-11	137
184	56	-35
147	-3	-8
2	-50	100
162	0	24
1756	-1	-1
25	-26	-12
49	2	28
15	8	0
0	/0	/0

Table 19: (geo.trn,geo.911) for 9/11/93

no. in ground truth	*%Hybrid*	*%Scheme F*
11	0	-10
3166	2	35
1939	-4	-57
22	-37	36
116	-8	-78
16	52	6
0	/0	/0
38	21	121
625	0	0
5	9	-60
28	5	17
7	-5	0
0	/0	/0

When averaging over the days, the test set will be representative and the previous conclusions of 15% to 20% errors hold. Individual days show a great deal of fluctuation. For instance for Class 4 the Hybrid percent errors are 9,-20,12,56,-11,-37, whereas the Scheme F percent errors are 161,112,200,96,260,137,36.

Training with 5/5 data

Table 20: (geo.55,geo.56) for 5/6

no. in ground truth	*%Hybrid*	*%Scheme F*
8	-6	0
4203	0	31
2762	-2	-45
26	-43	200
356	-2	-64
208	10	-5
0	/0	/0
164	14	38
1860	-1	0
29	0	-21
58	-7	8
15	6	6
0	/0	/0

Table 21: (geo.55,geo.57) for 5/7

no. in ground truth	*%Hybrid*	*%Scheme F*
5	-32	-20
2412	3	35
1676	-3	-43
33	-40	96
278	-22	-75
202	8	-5
0	/0	/0
84	10	38
1026	-1	0
9	0	-34
30	0	40
10	0	0
0	/0	/0

Table 22: (geo.55,geo.99) for 9/9

no. in ground truth	*%Hybrid*	*%Scheme F*
5	-52	-20
2126	0	29
1214	-6	-51
6	-15	266
95	50	-45

Table 22: (geo.55,geo.99) for 9/9

no. in ground truth	*%Hybrid*	*%Scheme F*
48	26	0
1	-100	200
88	1	18
782	-2	-1
6	16	-17
16	0	12
6	0	0
0	/0	/0

Table 23: (geo.55,geo.910) for 9/10

no. in ground truth	*%Hybrid*	*%Scheme F*
17	-24	-6
5081	4	33
3172	-12	-55
27	-48	137
184	78	-35
147	12	-8
2	-100	100
162	6	24
1756	-2	-1
25	-28	-12
49	0	28
15	13	0
0	/0	/0

Table 24: (geo.55,geo.911) for 9/11

no. in ground truth	*%Hybrid*	*%Scheme F*
11	-20	-10
3166	4	35
1939	-10	-57
22	-53	36
116	11	-78
16	101	6
0	/0	/0
38	33	121
625	-1	0
5	0	-60
28	3	17
7	0	0
0	/0	/0

Conclusion

Using representative training and testing data, the Hybrid method performs unquestionably better than Scheme F.

Even without representative training data, the better performance of the Hybrid method still occurs when using data of 5/5/93 as training: the improvement on Classes 4, 5, and 10 may be seen by a listing of the errors side by side on a daily basis. The Hybrid method did a factor of 3-4 better on Classes 8 and 11. Similarly, the Hybrid method gave almost an order of magnitude improvement on prediction of Classes 2 and 3 (at the 97% and 94% accuracy levels).

These results have occurred without adding any additional editing rules to the Class 6-13 editing procedure. Such rules are expected to improve the estimates on all the rest of the classes. For instance, Class 9 is predicted at a rounded-up 99% level accuracy by the Hybrid method but at the 100% level by Scheme F. An example of a rule would be to use the raw un-edited E1572(version 3.331) value for Class 9 that is also a rounded-up 100% accurate, but other rules might perform even better, were this of interest. Better performance could occur on Class 6 either by using the raw unedited E1572(version 3.331) value (96% accurate), additional rules, or by more data in the training sets (only 91 vehicles). The need for improvement on Class 12, already at the 96% accuracy level, is questionable.

Although the accuracy improvement is already extraordinary, the additional potential improvement achievable suggests that the Hybrid method needs additional study to clarify the issues on Classes 6-13 on a Class-by-Class basis.

Appendix A: An Event

Fig. 5 shows a rarely seen trailing wheel.

Appendix B: 9/9 As Training Set

Now for completeness we use 9/9 to predict the September data:

Table 25: (geo.99,geo.910) for 9/10

no. in ground truth	*%Hybrid*	*%Scheme F*
17	8	-6
5081	3	33
3172	-6	-55
27	-49	137
184	22	-35
147	-9	-8
2	-50	100
162	4	24
1756	-1	-1
25	-39	-12
49	0	28
15	13	0
0	/0	/0

Table 26: (geo.99,geo.911) for 9/11

no. in ground truth	*%Hybrid*	*%Scheme F*
11	9	-10
3166	4	35
1939	-6	-57
22	-63	36
116	-33	-78
16	56	6
0	/0	/0
38	25	121
625	0	0
5	-15	-60
28	3	17
7	0	0
0	/0	/0

Figure 3. An A2*4 vehicle classified as A6 and thus FHWA single unit Class 7

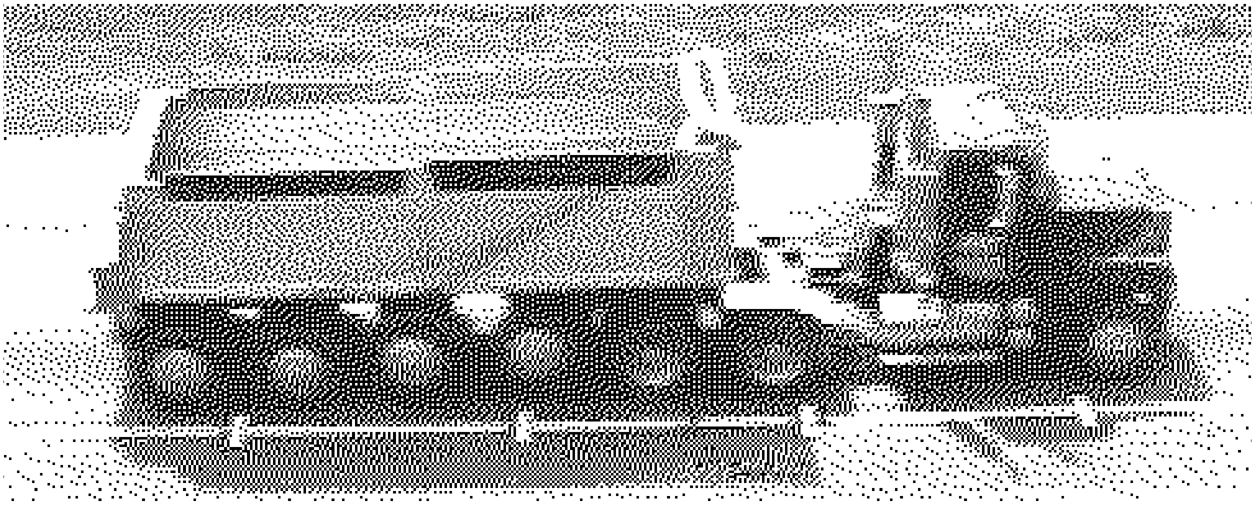


Figure 4. A B2 single unit vehicle. Or is it a two unit A1*2?

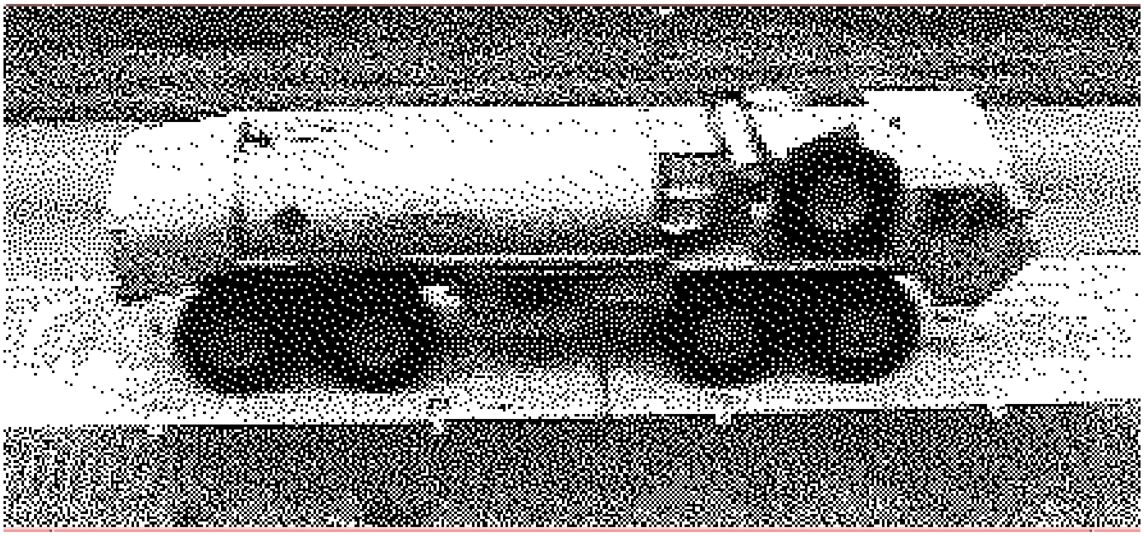
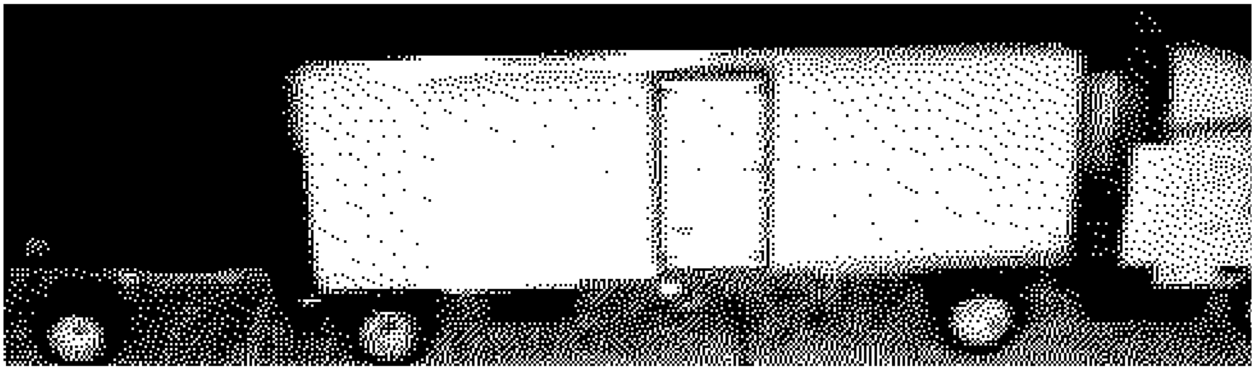


Figure 5. An A1*11 vehicle ready to tow another semi-trailer.



8/96

Fault Sensing Overview

After a background discussion, we will concentrate on the determination of classification faults based purely on measured vendor axle-separation values. The approach here represents what could be done with axle separation data currently being collected at many sites throughout the nation. How can we achieve automatic notification of faults (by flagging the data) whatever their cause, be it excessive electronic drift of equipment or power shortage without direct access to the private vendor electronics and algorithms?

Background

Classification data is used in a number of areas including safety, environment, and pavement management systems. In the USA, classification is often into one of the 13 FHWA classes of vehicles based on axle configuration, tires per half axle, and vehicle use. In Australia, a separate classification system is used for the purpose of pavement maintenance management. Weigh-in-motion data from the extensive CULWAY system [see Update of Traffic Design Chapter in the Austroads Pavement Design Guide- Status Report, Chris Koniditsiotis, June 1996] is combined with classification into axle groups such as tandem axles (pair of axles located near each other) with dual tires (TADT) or tridem axles (triples of axles) with dual tires (TRDT) rather than into vehicle types.

Faults can occur in a number of ways in vendor vehicle classification equipment used in the USA. Many of these conditions can, in principle, be determined at the time they occur in the equipment. The process of classification in most conventional equipment can be analyzed as several steps.

(1) Data acquisition: Road sensors to digital data.
(2) Sensor purification: Processing digital data by screening out part of the noise. This could involve assigning a fuzzy membership in both the noise and pure event classes. Our previous approach to equipment noise was to monitor the drifting noise level and automatically adjust the signal level required for event detection based on the noise level. We rejected the many very- low noise pulses and included the marginal events with amplitude flags. We also developed code to check either the logarithmic mean of key variables or the mean of the variable itself. When these exceed given limits, automatic notification occurs. The combined approach easily generalizes to a fuzzy membership algorithm.
(3) Vehicle parsing: the choosing of a set of digital sensor events to represent a vehicle event. This involves methods of dealing with extra and missing or fuzzy sensor events. This process is choosing one from a number of possible forms for the data. The form most consistent with the data is usually taken to be the selected form. Fuzzy form identities could also be used to improve precision.
(4) Invariant feature determination; e.g. axle spacing or vehicle length.
(5) Classification from invariant features. In general, this process could involve fuzzy alternatives and separate predictions from missing data.

Faults can occur in any of the five steps listed.
In steps (1) and (2), separate electronic and digital processing could determine types of noise present and signal-to-noise levels. One persistent fault that can occur in any of the five steps is a software or signal processing bug. Steps (1), (2) and (3) provide the vendor with possibilities for determining the internal consistency of the data. Also feedback from step (5) can provide additional checks. Although this internal information is not generally available to the user of commercial traffic classification equipment today, it could be in the near future. How do these considerations translate into traffic classification fault sensing from the data alone?

Three fault categories

There are three approximately-hierarchical categories of faults: (1) vehicle counting, (2) axle counting and (3) classification. These can be intermittent or persistent. Persistent problems that occur frequently are relatively easy to diagnose and are often associated with installation of equipment. Infrequent persistent problems have the same characteristics as intermittent problems. If they occur infrequently enough, they can be ignored. An example of the latter is the error found in some data occasionally having time event stamps with the units of seconds greater than 59. Fortunately, this error does not result in direct classification problems, but creates difficulties aligning the data with the ground truth determination, making it act as if it were a classification error.

The first category is associated with vehicle parsing or, in some cases, axle counting. When classification is involved, there are usually two or more separate sensor events that are combined into one number. The equipment tested in the Georgia I20 study [Georgia Department of Transportation Research Project 9210 Final Report] had a variety of precisions for vehicle counts.

Table 27: 7 day test accuracy; vendor. L indicates a Loop, P a piezo, and PR a resistive piezo sensor.

vendor identity by down stream location	*% (actual/vendor):Class 9 % (actual/vendor) and sensor types*
200’	114:115 L-P-L (Scheme F)
300’	104:118 P-L-P
300’+	136:114 P-L-P
300’++	124:104 P-L-P-(L?)
400’	88.5: 188(for data not lost) P-L-P
400’+	100:101 L-P-L
550’	105:/0 P-L-P (no Class 9)
600’	209:123 P-P
650	103:131 PR-L-PR
650’+	104:130 P-L-P
750’	106: 131 P-L-P
850’	123: 117 P-L-P

Comments in the report suggest that sensor performance can be good for vehicles detected but not good overall because of lack of sensitivity.Related phenomena are that Classes 2-3 can be very inaccurate while Class 9 is much more accurate. These are apparently effects of low signal to noise.

Given the correct axle separations, Class 9 (the ubiquitous 18 wheelers) can be a good indicator of the equipment’s ability to measure axle separations. The Scheme F results are better than 99% accurate with correct ground truth. If the vendors used a scheme that included Scheme F for Class 9, the results would be a measure of how well all axle separations are measured. A related phenomena is a dropped axle in a tandem pair.

Impact of dropped axles

One might expect dropped axles to account for all the Class 9 errors. Characterization of dropped axles by a

.

Table 28: First 48 hour test

axle/vendor	*[Class 9]-1.0*	*Vendor*	*Axle errors*
.13	.15	8476	1106
.04	.16	16120	598
.05	.14	11567	621
-	-	-	(300'++)
-	-	-	(400')
.01	.01	19476	265
-	-	6543	499
.03	.23	8915	235
.08	.31	15907	1236
.07	.30	18238	1217
.07	.31	20654	1444
.07	.31	18568	610

percentage of classified vehicles might be expected to track the Class 9 errors. If all axles were equally likely to be dropped, and if the likelihood of dropping a single axle were a small percentage, one would expect the Class 9 errors to be five times the dropped axle rate, because of the five axles. Other errors such as from the classification algorithm could make the Class 9 error rate higher. In Table 28 we show the dropped axle percentage in column 1 tabulated against 100 x (actual-vendor) /vendor in column 2; column 2 of Table 28 is related to the number following the colon in Table 27. Columns 3 and 4 of Table 28 are the raw vendor and dropped axle numbers taken from the GTRI study that give rise to column 1.

If one removes the first line of data, the rest straddle a line that has a classification error for Class 9 roughly three to four times the dropped axle percentage. Although this lends credence to a model that has dropped axles as the cause for Class 9 errors, the case is very muddy. The one line of data left out is the only one that is known to use Scheme F for Class 9; it has far fewer classification errors, 15%, than the expected 40-50%.

The rest of the data uses a default configuration of the manufacture. So it could be like comparing apples and oranges. Issues that need to be addressed to resolve these complexities are: (1) what are the dropped axle errors for Class 9 vehicles and how do they vary with axle number (2) If one uniformly uses Scheme F as the vendors' classification scheme, what are the Class 9 errors? (3) Are there other errors such as incorrect separation determinations that lead to incorrect assignment to Class 9?

Although these are interesting questions, our focus now is how to diagnose data faults using only the vendor data. Here we have already established that diagnosing dropped axles is important in determining fault counts for Class 9 and presumably other classes. In the case of substantially different loads on axles, dropped axles have a special case of *low sensitivity*. They could also have a case of *insufficient processing power* for a many-axle vehicle with closely spaced axles.

Low sensitivity

The data also suggest that dropped axles do not account for all the errors of classification. Of some 19-20,000 events possible. In some cases only 9000 were observed by the vendor. A possible mechanism here is low sensitivity of the detectors.

Low sensitivity is often an error of the persistent variety. For the user, it should show up by comparison to historical or calibration counts. A symptom is a higher than normal Class 9 fraction of vehicles. The manufacturer can diagnose this problem by examining the signal/noise ratio for each of the signals as described previously. In related fashion, the number of one-axle vehicles, or vehicles with only 3 of 4 expected axle registrations for P-L-P configurations are also indicators. For loops, a good signal in one of the two loops marks the condition.

Data lulls

A condition which is relatively easy to diagnose is the absence of data. Almost-absent data are more difficult to detect. Causes of almost-absent data such as congestion can be dealt with by examining speed or occupancy data. Road closures, bad weather conditions, or power failure could possibly be confused. The vendor has a slightly better grasp on the problem of nearly absent data by being able to say that all sensors either are or are not consistent with each other.

Rare events

Another way to determine faults is by the presence of a plethora of otherwise rare events. In theory, this condition can be diagnosed by carving events up into mutually exclusive parts of event space. Then by knowing the mean and variance of expected data, the presence of unusual data can be determined. One problem in practice with this type of method is for events that do not occur at all during training. How many of these events need to show up in testing before action is taken? Another practical problem is the determination of the expected baseline distribution. How does this distribution change with time and location?

A Class 9 Test?

The suggestion that dropped axle signals may not be an independent phenomenon will be investigated for Class 9. What axles are dropped for vendor vehicles with an exact time match to the ground truth vehicles? Can the answer lead to a metric for determining sensitivity of detection from vendor output data?

Class 9, the 18 wheeler class, should be able to be identified (to better than 99% accuracy) with good axle separation measurements. Yet up to 10%-88% identification errors occur with vendor equipment. Why?

The goal of this Class 9 work is to obtain an understanding of the types of problems that can occur, to be in a position to put next month’s work in context. One question, for instance, is how important are dropped axles in this class? If they dominate, they give an entirely different picture than if nonsensical axle separations data exists for these incorrectly classified vehicles.

9/96

Dropped Axles for Class 9

A portion of the vendor files were examined for dropped axle conditions to see whether or not they occurred randomly. Three vendors were chosen labeled vendor 1, 2, and 3, who had sites at downstream locations of 90m (300'), 200m(650'), and 260m (850').

The dropped axles were found to strongly deviate from a random error model. Vendor 1 showed a decided tendency to drop the first axle if only one axle was dropped. If the vendor had 2 dropped axles, the most likely case was that the axle separations that remained bore no perceivable relationship to the geometry of the ground truth vehicle. This we indicate by the letters *unc* for the uncorrelated condition. The next most likely condition for vendor 1 was a dropping of the last pair of axle hits. Another dropped condition was the first pair of axles, i.e. the first pair of axle hits. An axle hit represents an axle sensor event pair that detects the time of an axle crossing a pair of sensors at known downstream locations.

The other two vendors experienced less of a problem with the first axle hit. All vendors had substantial uncorrelated values. These are believed to be attributable to either lack of sensitivity of the equipment, or excessive noise in the environment. The latter could be of a highly intermittent nature, a characteristic shared with the quality of the collected data.

These results suggest several tests to detect these conditions. Many are covered by the neural network fault sensing method and because of that technique, we will not dwell on these possibilities. One example is a detector for two pairs of tandem axles corresponding to the case of a dropped first axle. The rest may be gleaned from studying the following results. The presence or absence of particular vehicles such as motorcycles would affect the utility of particular methods.

Vendor 1

1 axle missing

axle 1:44
axle 2: 0
axle 3: 2
axle 4: 3
axle 5: 3
unc : 6

2 axles missing

1&2: 9
1&4: 2
1&5: 3
2&5: 1
4&5: 13
unc: 16

3 axles missing

3&4&5: 3
1&4&5 or 1&2&3: 36

1&2&3: +2 more
2&4&5: 2
1&2&4: 1
unc: 13

Vendor 2

1 axle missing

1: 1
2: 3
3: 4
4: 0
5: 1
unc: 1

2 axles missing

unc: 7
4&5: 2

3 axles missing

unc: 34
3&4&5: 17
2&4&5: 4
1&2&3: 2 plus below
1&4&5 or 1&2&3: 1

Vendor 3

Vendor 3:
1 axle missing

1: 14
2: 2
3: 1
4: 16
5: 15
unc: 23

2 axles missing

4&5: 1
unc: 7
1&4: 1
2&5: 6

3 axles missing

unc: 26
1&4&5 or 1&2&3: 2
3&4&5: 8

2&4&5: 1

Separation Fault Sensing

The first of two methods for sensing faults in axle separation measurements is a go/no-go test for each separation. These are presented irrespective of the numbers of axles on a vehicle. Only those separations present should, of course, be tested. There could be utility in refining these limits to being dependent on the number of axles detected, but such improvements are expected to be small compared to the second method.

Axle separations out of bounds

1	4.0	30.5
2	2.7	63
3	1.8	52
4	2.2	43
5	2.2	31
6	3.5	18

These axle separation faults are plotted for the three vendors in graphs entitled *separations out of bounds*. The connection to make is that these values strongly correlate the neural net (NN) results labeled *NN rejects for all classes.*

NN rejects

A generalization of the concept of axle separations being out of bounds is that of *rare vehicles*. Plots of such vehicles are given in Figs. 6-8. If, for instance, there was no vehicle present in an extensive data base having every axle separation value within 30 cm (1') of the test vehicle under consideration, we could label that vehicle as rare and include it in the reject count. Two-axle vehicles are seldom rare by this definition. In fact, if we look at the plots for the three vendors labeled *All rejects by class* we see that they are predominately Class 9 vehicles.

The criteria

Of all the vehicles shown in the *Hourly vehicle count* graphs some of them are classified correctly and some are not. In order to determine a correct classification a correspondence between the vendor vehicle and a ground truth vehicle must be made. This is usually done on the basis of the registered time of the vehicle. We have discussed this topic extensively in previous Traffic Monitoring Progress reports. Certain of the vehicles cannot be matched. A count of these is shown in the graphs labeled *Unmatched vehicles*. A large number of unmatched vehicles early or late may indicate different start and stop times for the vendor data set and the ground truth data set.

What we are trying to determine is the occurrence of classifications that disagree with the ground truth classification. Because the Class 9 ground truth vehicles are classified better than 99% using Scheme F for the GTRI data set of 1993, we determine the hourly differences between Scheme F determinations by the vendor and the ground truth determination by the same scheme. This approach removes the vendor's method of classification as contributing to the error we measure. The number of different classifications are shown in the graphs labeled *Scheme F differences. T*he majority of difference are from Class 9 as the graphs labeled *Scheme F Class 9 only* show.

The rejected vehicles from the NN are labeled *reject vehicles for all classes*. The difference in the numbers of these is shown in the first two such graphs for the first vendor. The first of those graphs uses 1/3 of the GTRI 48-hour ground truth data set. The second uses all of it. The rest of such graphs use all of the GTRI 48-hour data.

The rest of the graphs show the total Class 9 traffic, the part of that traffic that is rejected by the NN because of the wrong number of axles, dropped or added, and the part that arises from different values of axle separations. The criterion to be included in the count is that the sum of the absolute values of the difference in separations be larger than 2.4m (8').

The question of believability of these differences is addressed in the comparison between sections of a vendor data base and the corresponding ground data base during the height of the disagreements in classification.

First vendor results

Fig. 6a Hourly vehicle count

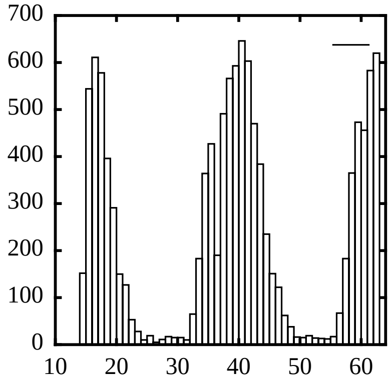


Fig. 6b Unmatched vehicles

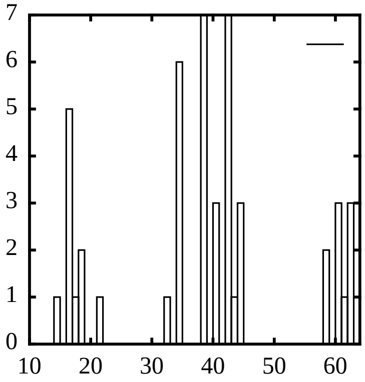


Fig. 6c Scheme F differences

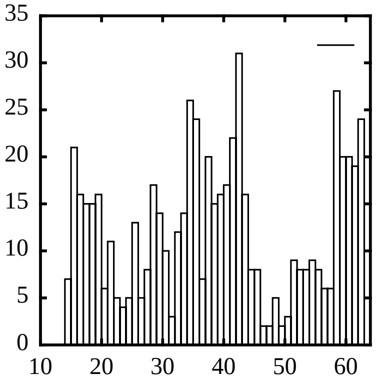


Fig. 6d Scheme F Class 9 only

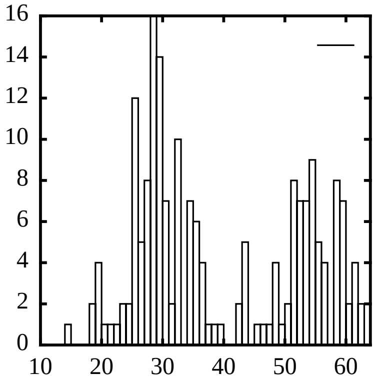
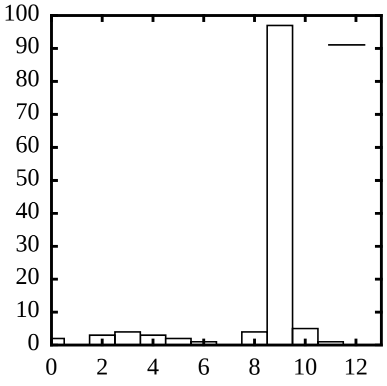
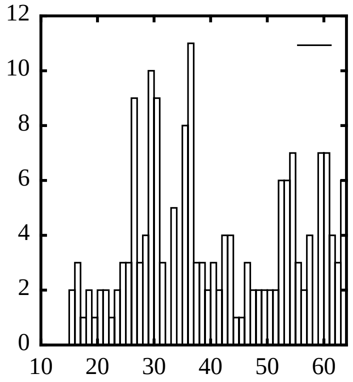


Fig. 6e All rejects by class



PNN with training set only

Fig. 6f NN rejects for all classes
1/3 georgia data training



PNN with all georgia.txt data, for training (the results below use the same).

Fig. 6g NN rejects for all classes
using all georgia data

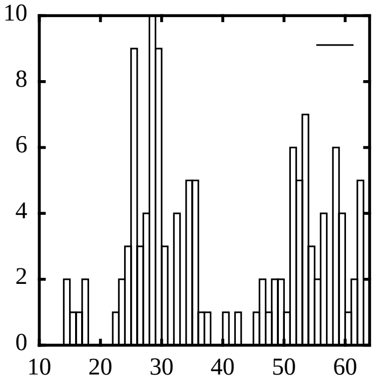


Fig. 6h Separations out of bounds

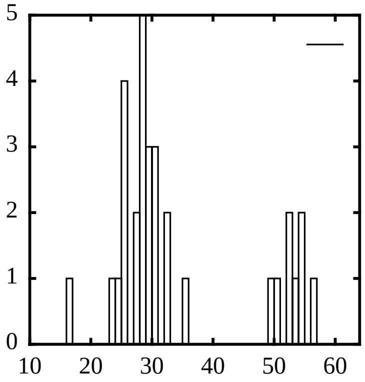


Fig. 6i Total Class 9 hourly

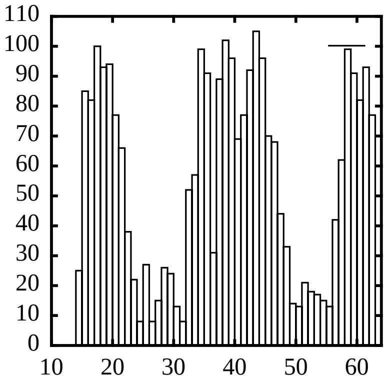


Fig. 6j Rejected Class 9 with wrong number of axles

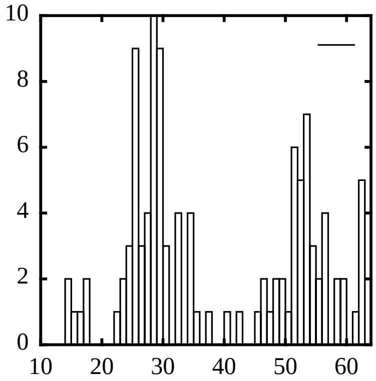
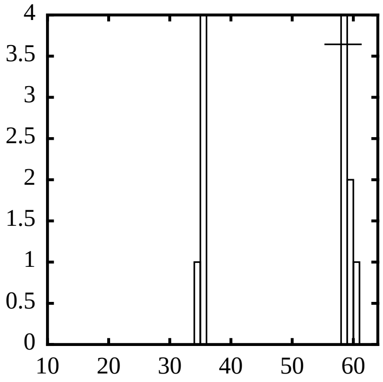


Fig. 6k Rejected Class 9 with very different spacings



No speeds > 75 m.p.h. for rejected Class 9 vehicles.

Second vendor results

Fig. 7a Hourly vehicle count

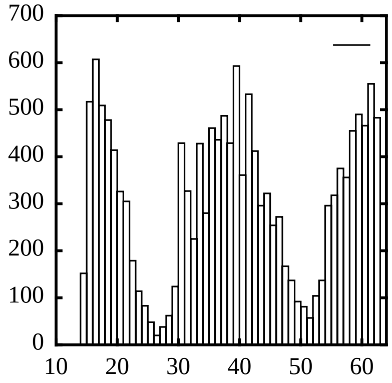


Fig. 7b Unmatched vehicles

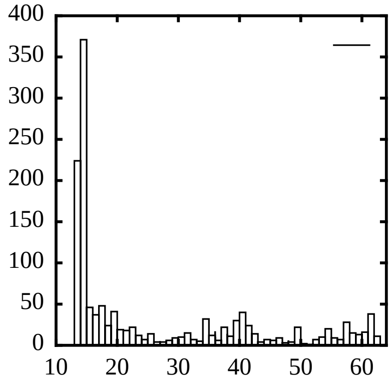


Fig. 7c Scheme F differences

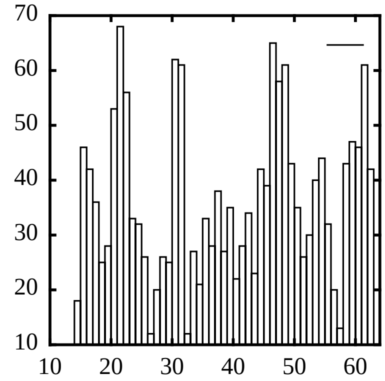


Fig. 7d Scheme F Class 9 only

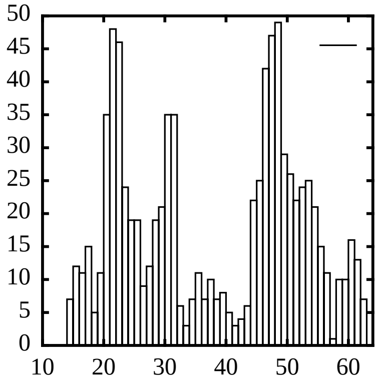


Fig. 7e All rejects by class

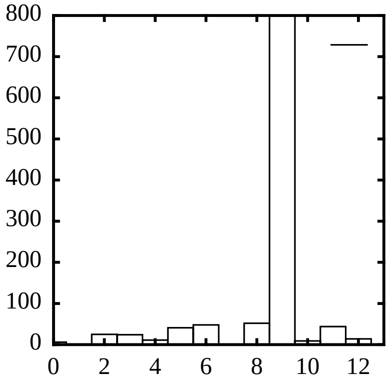


Fig. 7f NN rejects for all classes

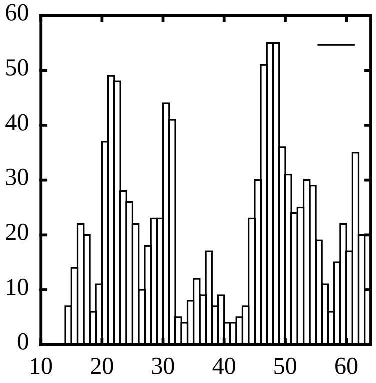


Fig. 7g Separations out of bounds

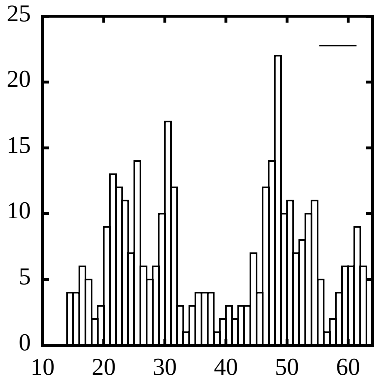


Fig. 7h Total Class 9 hourly

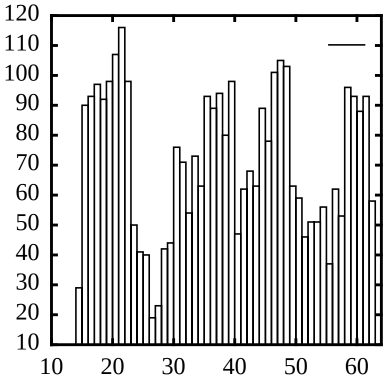


Fig. 7i Rejected Class 9 with wrong number of axles

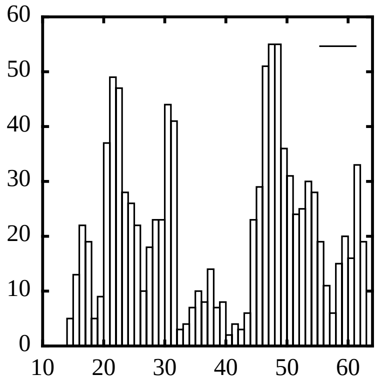
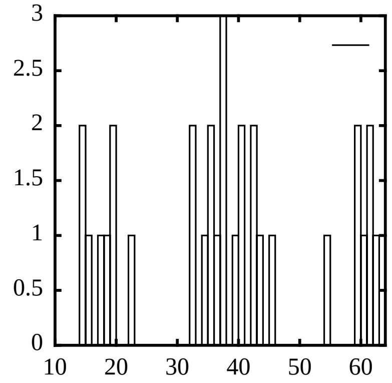


Fig. 7j Rejected Class 9 with very different spacings



No speeds > 75 m.p.h. for rejected Class 9 vehicles.

Third vendor results

Fig. 8a Hourly vehicle count

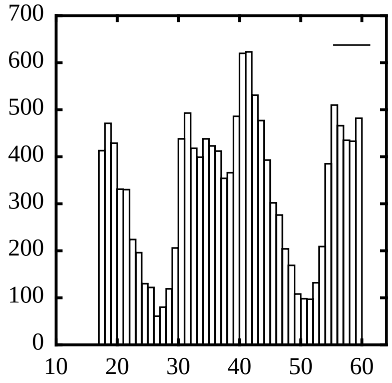


Fig. 8b Unmatched vehicles

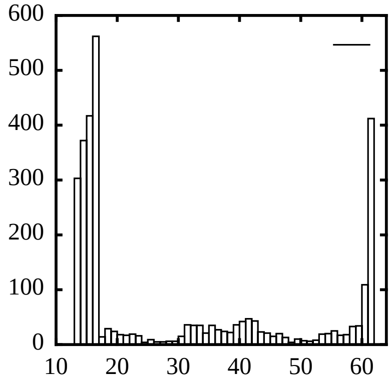


Fig. 8c Scheme F differences

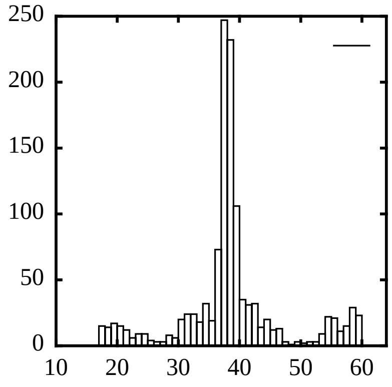


Fig. 8d Scheme F Class 9 only

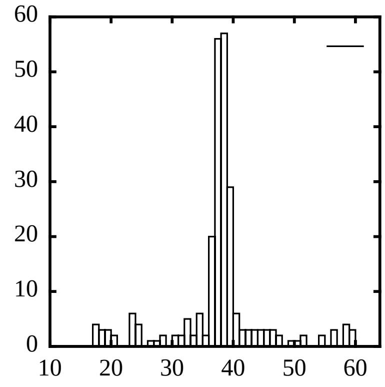


Fig. 8e All rejects by class

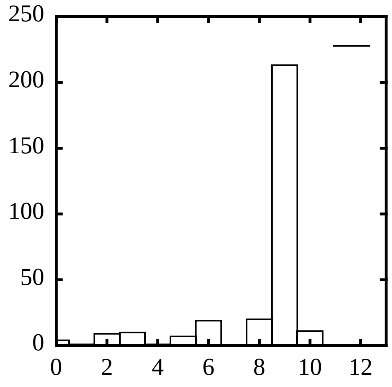


Fig. 8f NN rejects for all classes

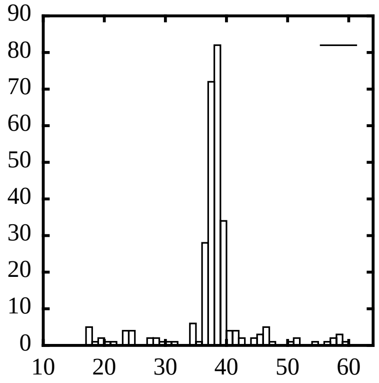


Fig. 8g Separations out of bounds

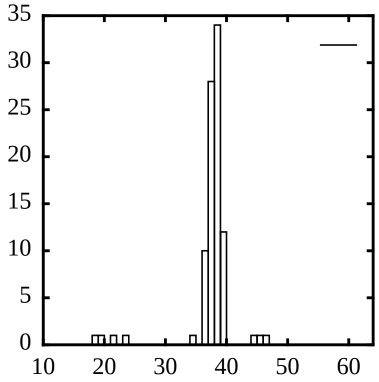


Fig. 8h Total Class 9 hourly

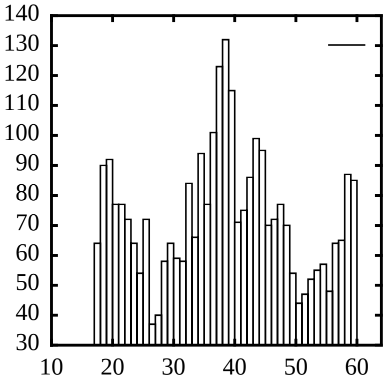


Fig. 8i Rejected Class 9 with wrong number of axles

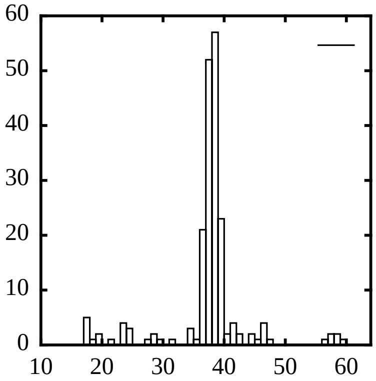


Fig. 8j Rejected Class 9 with very different spacings

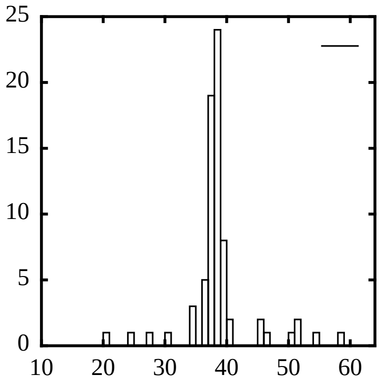
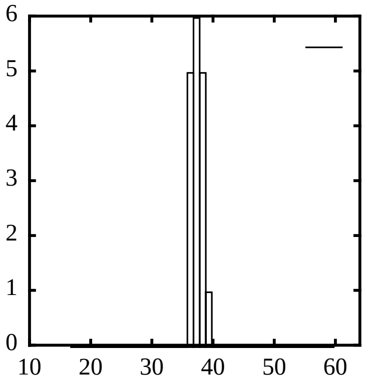


Fig. 8k Speeds > 75 m.p.h. and rejected Class 9



The Vendor 3 entries near 8984 (gt 9599)
(vendor id), ground truth id, date, time in hhmmss, classification, (speed), wheel-base, axles, axles-1 separations. xxxxxxxxxxxx labels corresponding entries.

```
8979 9588 5-6-93 150810 5 68.5 15.3 2 15.3
8980 9592 5-6-93 150852 2 41.7 21.7 4 3.8 15 2.9
8981 9594 5-6-93 150902 3 63.5 14.4 2 14.4
8982 9595 5-6-93 150906 2 69.1 9.4 2 9.4
8983 9596 5-6-93 150912 5 63.5 17.5 2 17.5
8984 9599 5-6-93 150925 9 43.4 31.8 5 5 2.9 20.7 3.2   xxxxxxxxxxxxxxxxxxxxxxxx
8985 9604 5-6-93 150956 2 27.1 7.7 2 7.7
8986 9605 5-6-93 151002 1 33.3 3.7 2 3.7
8987 9606 5-6-93 151005 2 57.8 7.8 2 7.8
8988 9607 5-6-93 151007 2 56 6.3 2 6.3
8989 9608 5-6-93 151014 2 24.2 7.7 2 7.7
8990 9609 5-6-93 151016 2 59.1 8.4 2 8.4
8991 9610 5-6-93 151018 3 73.5 11.2 2 11.2
```

The ground truth entries near 9599 -- no speed or vendor id (lch indicates lane change)

```
9590 5-6-93 150822 3 10.55 2 10.55
9591 5-6-93 150834 2 9.20 2 9.20
9592 5-6-93 150852 9 42.69 5 10.31 4.14 24.38 3.86
9593 5-6-93 150858 3 10.65 2 10.65
9594 5-6-93 150902 2 7.97 2 7.97
9595 5-6-93 150906 3 8.80 2 8.80
9596 5-6-93 150912 3 10.53 2 10.53
9597 5-6-93 150913 2 8.61 2 8.61       lch
9598 5-6-93 150917 2 8.26 2 8.26
9599 5-6-93 150925 9 48.08 5 10.51 4.39 29.34 3.84     xxxxxxxxxxxxxxxxxxxxxxxx
9600 5-6-93 150927 3 9.49 2 9.49
9601 5-6-93 150933 2 8.32 2 8.32
9602 5-6-93 150938 2 8.24 2 8.24
9603 5-6-93 150951 2 9.67 2 9.67
9604 5-6-93 150956 5 18.63 2 18.63
9605 5-6-93 151003 6 22.86 3 18.54 4.32
9606 5-6-93 151005 3 9.19 2 9.19
9607 5-6-93 151008 2 7.64 2 7.64
9608 5-6-93 151014 5 20.04 2 20.04
9609 5-6-93 151017 2 8.70 2 8.70
```

Conclusions

Most of the detectable reject events correlate well with the events having out-of-bounds axle separations. These vehicles are predominately Class 9 vehicles. They also correlate well with those vehicles having been given the wrong Scheme F designation using the vendor’s values of the axle separation.

The differences between the Class 9 rejects with wrong numbers of axles and the total Scheme F errors are primarily two-axle vehicles, which are particularly resistant to detection of wrong values.

If the conditions present during the GTRI study are representative of typical vendor installations, it appears feasible to specify the reject vehicles as a measure of the performance of the vendor equipment.

This metric can be improved by obtaining more extensive ground truth sets of verified axle separation measurements. It does not suffer from the problem of generalization that a method based on the individual distribution of events in axle separation space would have.

If the underlying mechanism is poor signal to noise ratio, then we could expect that Class 9 is not special in the effects of the noise on the data. It is simply the Class for which effects of incorrect separations can be most easily observed.

The intermittent nature of the reject vehicles is especially prominent in the Vendor 3 data. There, around the 38’th hour, something has caused a change in the performance of the equipment. This vendor also shows extensive speed errors during this time period.

If we hypothesize a radio frequency or line current noise source, it would have to be very local because of the differences to the vendors located within hundreds of feet of each other. It would also have to account for the sensitivity of Vendor 1 to errors at night, but not so much to those during the day.

Once a metric for measuring the performance of vendor equipment is established, there is then the hope that the conditions under which the performance is better or worse can be determined.

One way of presenting the same signal-to-noise environment to equipment by various vendors is to do so in a test center where signals are simulated and controlled.

It may well be that some vendors have special filters to remove kinds of noise but not others. Their equipment could perform differently in different environments. The noise environment of a shopping center is a special challenge where large spikes associated with changes in lighting can occur.

These issues can be investigated with current technology, and successful addressing of them could pave the way for dramatic improvement in the traffic statistics that underlies the hope of the ISTEA legislation. Better statistics are the foundation for workable pavement management and more realistic economic, environmental, and safety projections.

Extracted Potential-Classification-Schemes Articles

The following articles have been extracted from progress reports from issues of Traffic Monitoring Progress (TMP) 6/95 to 12/96. They serve as a complete reference to TMP articles on the subject of the Potential Classification Schemes, giving details of items mentioned in the In-Depth Review of Potential Classification Schemes. The articles appear in chronological order. Minor editing has occurred including renumbering figures and tables, providing uniform titles, and clarification of context.

7/95

AC Code Modifier Code

The modifier code of the AC code standard E1572 is specified in a Fortran program as a part of the standard. It plays a key role in translating the code to the FHWA 13 class system.

The modifier is based on the axle spacings, and some concepts based on axle groups. An axle group is one or more axles determined to belong together to represent the concepts of: isolated axle, tandem axle (a pair of closely spaced axles), tridem axle (triple axles), and quad (four axles). The algorithm producing the segmentation into these groups is described in the standard and its details are beyond the scope of the current discussion. See article entitled *E1572 FORTRAN Algorithm* below.

Multiple axle groups are further characterized according to the axle separations within the group as being commercial or recreational. Any axle separation less than or equal to the distance RECMAX (recommended to be set at 1.1 m or 3.5’) is considered to be recreational. Axle separations greater than the recommended RECMAX distance are considered commercial. Other parameters in the modifier definition are:
UTLMAX = 6.1 m (20’)
PIKMAX = 4.3 m (14’)
TWOMAX = 6.1 m (20’).

To simplify the presentation we call a commercial group a *comdem* and a recreational group a *recdem*. Such groups have one or more axle spacings in its category. Thus an axle group can either be:
(i) single (isolated) (ii) recdem (iii) comdem or (iv) recdem and comdem.

After a vehicle is separated into units such as a tractor or a trailer, the axle groups are determined. There are two parts to the determination of the modifier code. One is based purely on the first axle separation and is intended for what is called non-commercial vehicles. (Closer examination of the algorithm shows that a better name might be single unit vehicles with or without non-commercial trailers). We label this method as *NC*. The second part is based on a what may be viewed as a decision tree. The concept of the distance between the last axle of the first unit and the first axle of the second unit we call the *draw*.

The NC category has parameters in addition to those above of:
STDMAX = 2.9 m (9.4’)
COMMAX = 2.6 m (8.5’)
MOTMAX = 1.8 m (6.0’).
The modifier code values are indicated after a colon for the NC decision tree:

```
sep <= MOTMAX: 1
sep <= COMMAX: 2
sep <= STDMAX: 3
sep <= PIKMAX: 4
sep <= TWOMAX: 5
else         : 6.
```

Here sep is the first axle separation.

The hierarchical decision tree for the modifier can be written as below; here alternatives at the same indentation level are taken one after the other until a match is reached indicated by the colon:

```
2-axle tractor vehicles
  1-unit vehicles    : NC
  2-unit vehicles
    1-group trailer
```

```
      comdem trailer : 9
      recdem trailer : NC
      draw < UTLMAX  : NC
      else          : 9
    multi-group trailer
      comdem trailer : 9
      recdem trailer : NC
      draw <= PIKMAX : NC
  >2-unit vehicles
    recdem(any) not
    comdem(any)      : NC
    else             : 9
>2-axle tractor vehicles
  1-unit vehicles
    comdem(any)
      single&tandem and
      first axle
      sep> TWOMAX    : 6
      else          : 7
    recdem (any)     : 5
  2-unit vehicles
    1-group trailer
      comdem tractor
        comdem trailer : 9
        recdem trialer : 7
        draw < ULTMAX  : 7
        else          : 9
      recdem tractor
        not comdem     : 5
        comdem         : 9
    multi-group trailer
      comdem tractor   : 9
      recdem tractor   : 5
  3 or more unit vehicles
    recdem & not comdem:NC
    else               : 9
```

An advantage of writing these rules as a decision tree is that one can spot incompleteness in the rules. In practice, the rules may not need to cover all possible ranges of the parameters such as the draw or the first axle separation. For instance, the above rule set does not give a modifier under the following conditions:
(1) 1-axle vehicle
(2) 2-axle tractor with a single multi-group trailer having draw > PIXMAK
(3) solo tractor with > 2 axles and not condem and not recdem
(4) tractor with > 2 axles with single trailer having a single axle
(5) tractor with > 2 axles with trailer having two isolated axles.

Of these only case (2) occurred in practice, on 31 occasions of the A1*11 vehicle in the Georgia data set e.g. separations of 15.65’,18.59’,8.37’. The repaired version of E1572 incorporating these changes is called E1572 (version 1.0).

Improving the AC code?

One of the questions we had answered in the negative until recently is whether there is sufficient data to judge the choice of parameters in the AC code. Viewing that code as having hidden parameters (the vehicle configuration) for which we have no ground truth data set, and evaluating the code solely on how well it classifies the Georgia data is the current approach. This is carried out by comparing the classification of the AC code with the probabilistic neural network (PNN) described at the NATDAC conference.

The AC code answers many of the objections to most if/then methods of classification. It constructs underlying variables that arrive at an intermediate truth, the vehicle structure. Then based on that, a classification scheme is derived. The modifier part of the AC code can itself be put into the form of a decision tree (see article entitled *AC Code Modifier Code* above). Can we improve this methodology?

The AC FORTRAN code we use was obtained from David Huft shortly after the standard had been approved as E1572. Although the code was written in FORTRAN, we linked it to two C functions by passing argument values by reference and by properly adding null characters to the end of strings. The C code has been linked to our C++ code in standard fashion. We applied this FORTRAN code (version 0) to the training set geo.trn, see Tables 3-5, consisting of a random choice of 1/3 of the entire GTRI data set.

Showing the true vehicle class on the vertical column and the different classes into which it was classified on the upper row, we have characterized the performance on the training set. We will use the training data as the basis of evaluating any change to the parameters of the AC code.

TABLE 1. E1572 (version 0) part 1

	1	*2*	*3*	*4*	*5*
1	14	1			
2		5, 204	1, 053		
3		1, 359	2, 573		37
4			3	30	3
5		2	147	53	164
6			1	1	
7					
8		9			
9			6		
10			1		
11					
12					
13					

TABLE 2. E1572 (version 0) part 2

	6	*7*	*8*	*9*	*10*
1					
2			3		
3	1		35		
4	5		1		
5	2		20	2	
6	233		1	6	

TABLE 2. E1572 (version 0) part 2

	6	*7*	*8*	*9*	*10*
7		1			
8			211		
9		1		2, 206	
10		1			28
11			2		
12					
13					

TABLE 3. E1572 (version 0) part 3

	11	*12*	*13*	*-*	*-.*
1					
2					
3					
4					
5					
6		1			
7					
8					
9					
10			6		
11	68				
12		21			
13			1		

Comparison with the PNN values gives rise to a number of conclusions about E1572(version 0):
(1) class 5 is too frequently misidentified as Class 4 (53 times versus 13 or 11)
(2) 5->8, occurs 20 times versus 2 or 3 times for the PNN.
(3) class 3 is too frequently misclassified as Class 8.
The rest of the comparison is quite favorable with respect to PNN: better on classes 11, 12 and 4 and requiring no reclassification scheme. There is an error greater than 100% of the ground truth on Class 4 that has the potential of being reduced substantially by correcting the 5->4 misidentification. The 20% error of Class 8 arises primarily from 3->8. The balance issues apply to inherently overlapped regions in (2) where it is believed the only remedy is data on additional measurements of vehicle properties.

The first question to answer in the three issues raised above is whether the problem is in the vehicle structure or the modifier determination. The structure consists of the number of physical units and a description of the number of axles in each axle group on each unit. It also indicates the beginning of a semi trailer unit by an * and a front supported vehicle unit by a capital letter, the ordinal number of which is the number of axles. The modifier indicates information extracted based on the structure and the axle spacing details. We attempt our classification process without a database for the structure of each vehicle. We must rely on the raw axle separations and overall vehicle length, although we can refer to the original video for verification of any puzzling situations.

The structure versus modifier question can be answered by looking at the AC code for the classes involved in the trade-offs. For (1), Class 5 and Class 4 are both primarily of A1 structure. Thus, the problem is apparently the modifier and not the structure. For (2), Class 10 and Class 13 the issue is the number of trailers. The Class 3 part arises from a confusion of recreational and commercial trailers. The issue is structural. For (3) When Class 3 is too frequently misidentified as Class 8, it is partially due to the case(2) incompleteness of the modifier code. This condition accounts for 4/35 misclassifications. The rest has to do with the recreational versus commercial trailer determination. The two PNN test sets of comparable size had 3 and 2 misclassifications into Class 8 from Class 3 versus the 35 we observe.

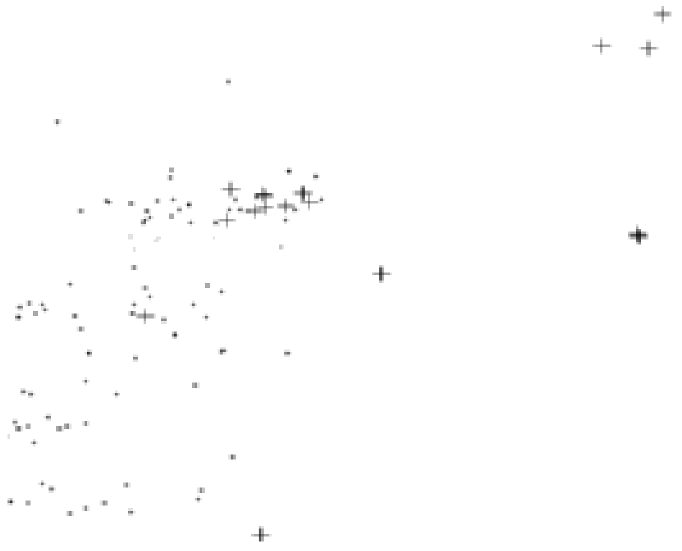
5->4

Lets start with the Class 5 misidentification into Class 4 problem. Because of the higher population of Class 5 vehicles, it is less of a problem for it than for Class 4. The key modifier parameter is TWOMAX. The vehicles that are being classified as modifier 6 and therefore Class 4, but are really Class 5 vehicles in the ground truth set, all but 3 have the first axle separation greater than 20’ and less than 22’. All of these vehicles have structure A1, i.e. single unit 2-axle vehicles.

With a value of 22’ for TWOMAX, the 53 misclassifications drop to 3 at the expense of a single new Class 6 vehicle being identified as a Class 4. The new value of TWOMAX looks like a reasonable compromise; it reduces the Class 4 off-diagonal errors as a fraction of the total Class 4 population from over 150% to about 60%. This modification is known as version 1.20.

Because of the large number of passengers on a Class 4 vehicle, the importance of the class is enhanced in determining passenger miles of travel. When this Class 4 information is required, it is desirable to reduce the 60% error. The number of classified Class 4 vehicles is an over-estimate of the actual population. However, editing the data by a simple edit matrix for the Class 4 and Class 5 populations will lead to erroneous results as can be seen in Fig. 1.

Figure 1. Overlap of Classes 4(+) and 5(.) plotted against vehicle length and wheel base using xgobi.



In Fig. 1 we see that where Class 4 and 5 Overlap, they are approximately of the same density, but much of the Class 5 region has no Class 4 vehicles. The data in this figure are plotted against the wheel base and the overall vehicle length. We can see that the latter does not contribute much to separating the data. A better approximation in editing the data is to define an additional region, a Class 4&5 region, where the data overlap. The best way to edit the data is to have the information based on bins for the axle separations. Each bin would then have a probability for belonging to each of the classes.

The overlap in the figure also shows us why there is not much hope of reducing the off-diagonal error-percentage in classification.

3->8 and 8->3

Class 3 being misidentified as Class 8 dilutes the true Class 8 content. The AC codes of the problem classifications are 18 A1*1, 13 A1*2, and 4 A1*11 structures. The A1*11 structures are associated with the lack of a complete modifier mentioned earlier. Correcting the modifier problem results in what we call version 1.0 of the AC code.

The three Class 8 vehicle structures are A1*1, A1*2, and A2*1. Of these 6 A1*2 and 3 A1*1 are misidentified as Class 3 (with a trailer). Treating a car or pickup-pulled trailer as a semi-trailer, both the A1*1 and the A1*2 structures are suitable for Class 3 and Class 8, and it appears that making the choice between Class 3 and Class 8 for each of those structures is the problem at hand.

First we concentrate on the 18+3 A1*1 vehicles. The key distinction between Classes 3 and 8 depends on the modifier. A commercial modifier means Class 8 and a NC modifier means Class 3. These vehicles have 2 axle tractors, 2 units, and a single group trailer. The A1*1 is not a condem or recdem, terms used in the AC code modifier article above. The entire A1*1 decision rests on the parameter UTLMAX (20’). With draw length (the 1*1 distance in A1*1) below ULTMAX, the vehicle is designated NC and therefore Class 3 (this also requires 9.4’ < sep1 < 22’ up from 20’ by our Class 4 change.) Above UTLMAX the vehicle is commercial and all such two-unit vehicle with three or more axles are Class 8.

Changing UTLMAX also affects another modifier choice of 7 or 9. However, this choice doesn’t affect the FHWA classification because both choices are commercial (>6), and the FHWA decision is then based on the number of axles and the number of units only.

Lets look at a few of the draw lengths of the A1*1 misidentified vehicles:

TABLE 4. A1*1 draw lengths

A1*1	21.27
20.2	20.05
21.13	20.83
20.66	21.73
21.28	20.84
20.87	21.65

TABLE 5. A1*2 draw lengths

A1*2	11.69
14.81	13.93
16.96	24.21
16.53	11.63

Raising ULTMAX from 20’ to 22’ can cure the classification problem for A1*1 3->8 but could induce additional problems. This parameter appears to have little to do with the A1*2 3->8 problem where most of the draws are lower. Let’s examine the effect of raising ULTMAX. The effect outside minor changes to Classes 2 and 3, which have big populations, is dropping the misidentification of 3->8 from 35 to 17 at the same time raising the misidentification of 8->3 from 9 to 23. This we indicate by the shorthand (9,35)->(23,17).

Although this situation appears to be a classical problem where two distributions overlap (not unlike passenger cars and pickups) where making one misidentification better makes the inverse process worse, the situation is actually more complicated. The (23,17) misidentification numbers for 8->3 and 3->8 respectively are still large compared to the PNN on the test data with (10,7) and (7,6). The 23 8->3 misidentifications with ULTMAX = 22’ have structures: 6 A1*2 and 17 A1*1. The 17 3->8 errors are divided between 13 A1*2 and 4 A1*11. Almost all of the A1*1 errors have been transferred from (3,18) to (17,0) with the larger ULTMAX. The 18 A1*1 misidentification problem was cured at the expense of the inverse problem. The A1*2 A1*11 numbers didn’t change at all. The parameter ULTMAX set at a value of 21’ gives a balanced error of (6,7) for the A1*1 vehicles alone. Invoking this value in Version 1.0 of the AC code is using version 1.10 of the AC code.

A1*2 3->8 and 8->3

Next we turn to the A1*2 semi-trailers. These have draws of 11’ to 24’ most being under 17’. The modifier code determines the classification based on the vehicle being comdem (Class 8) or recdem (Class 3). Because there is a multi-axle group on the trailer, if the tandem spacing is not greater than RECMAX = 3.5’ the vehicle is Class 3, otherwise it is Class 8.

To study this case we take all the Class 3 and Class 8 vehicles that have structure A1*2 and produce a data file for the version of Breiman’s decision tree algorithm in the IND package called CART with 41 samples for pruning and 200 samples for training. Three axle separations are involved in this data set: SEP1, SEP2, and SEP3.

The expected decision on the draw length (SEP2) will arise because the 8->3 misidentifications occur on vehicles with SEP2,SEP3 values of
34,3.3
32,3.45
31,2.75
31,2.80
These are typically the low, wooden-bed, open flatbeds.
Some 3->8 misidentifications are:
20,7.8
17,4.72
27,5.91
12,7.73
15,4.05
17,7.75
Three examples of these are: a pickup pulling a trailer carrying three lightweight rolls of what looks like insulation about a meter in diameter each; a pickup pulling an empty wooden-flatbed trailer; and a small mobile home pulling a car.

These axle spacing values suggest that above a 30’ draw we should automatically consider the vehicle to be Class 8. These conclusions, though with more precise parameters, are just what the decision tree arrives at in a more systematic way also using the 3->3 and 8->8 A1*2 data. This CART algorithm produced the following decision tree involving only the draw length and the tandem spacing.

```
sep2 < 22.785:
|   sep2 < 18.23:    C3
|   sep2 >= 18.23:
|   |   sep3 < 3.41: C3
|   |   sep3 >= 3.41:C8
sep2 >= 22.785:     C8
```

The sep3 critical value of 3.41' suggests lowering the RECMAX value for A1*2 vehicles. Whether lowering to this value also benefits the other recdem versus comdem decisions awaits a test case.

What apparently should be modified is the part of the modifier tree given above to read

```
2-axle tractor vehicles
  1-unit vehicles : NC
  2-unit vehicles
    1-group trailer
     tandem group
      draw < 23
        draw < 18:      NC
        tandem < 3.4:   NC
        else:           9
      else:             9
     non-tandem group
      comdem trailer : 9
      recdem trailer : NC
      draw < UTLMAX  : NC
      else           : 9
```

Incorporating this scheme reduces the A1*2 errors from (12,24) to (7,16), now just over 10% of the Class 8 population. There is another 10% error from 5->8. This modification of ULTMAX results in version 1.10 which with the change in version 1.20 is known as version 1.30.

Including the 5->8 errors

We can look at the effect of including the 5->8 errors in the decision tree simultaneous with the 3->8 and 8->3. The tree class now has two elements NC and C (modifier 9).

```
sep2 < 24.64:
|   sep3 < 3.745:  NC
|   sep3 >= 3.745:
|   |   sep2 < 21.21:  NC
|   |   sep2 >= 21.21:  C
sep2 >= 24.64:  C
```

The order of importance of the variables is switched in order from the previous decision tree. The first decision is still that which separates out the long wooden flatbeds. The second decision is now concerned with the tandem spacing. Now, the wide tandems can be either NC or C. Here Class 5 is called NC too.

The performance of the new arrangement is (8,13+18) versus the previous (7,16+21).
Although the parameters appear to have a different order of importance in the decision tree, the decision tree above can be put into the same form as the previous tree with only parameter changes of
22.75 -> 24.64,
18.23 -> 21.21,
3.41 -> 3.745.
These changes represent a different compromise based on using the Class 5 vehicles together with the Class 3. The total off-diagonal error as a percentage of all Class 8 vehicles has been reduced from 25%

to 18% and this value compares to 15% and 13% for the two test cases of the PNN. If we remove the 4 A1*11 vehicles which we argued we should do, the 18% drops to 16% and is closer to the PNN value. This decision tree is incorporated in Version 1.30 becomes Version 2.31 of the AC code

To see if we can get additional performance, we can attempt to build a decision tree for the A1*1 vehicles. The tree for Classes 3, 5 and 8 is

```
sep2 < 20.895:  NC
sep2 >= 20.895:  C
```

This critical value is very close to the value of 21 we chose for ULTMAX previously that we keep.

The A1*2 classification patch below becomes Version 2.33 of the AC code. At end of section of 2-axle, 2 unit vehicles just before >2-unit vehicles:

change PIKMAX to 20' and with further additions the decision tree segment is

```
draw <= 20'  : NC
spread < 10' : NC
else         : C  ( mcode = '9')
```

Other arrangements differing from the above may also work well and could be consistent with a more general underlying rule. A generalization of what we have done is to create a separate decision tree for each vehicle structure. How much that would improve the classification remains to be seen. However, with these changes, we are getting very close to the PNN performance level on Classes 6-13.

E1572 FORTRAN Algorithm

The first part of the FORTRAN algorithm is determining the axle groups. The next step uses inter-group information in the unit identification. The structure code follows from the units and groups. It serves as the basis of the modifier number determination and the closely related FHWA classification described earlier.

Definition of a group

A group has an axle count, and if it has a count greater than unity, it has a non-zero average length. An axle is either added to an existing group or made to start a new group. Any candidate for an existing group must be less than a fixed distance known as the group limit = 8.0’. If it meets this requirement and there is a count of unity in the group it is admitted; if there are already two or more members, it has to meet an additional conformity condition. The discrepancy between an axle and an existing group is the difference in separation to the nearest axle in the group and the average length of the group. If the discrepancy is less than a fixed number known as the group delta = 2.0’, it is admitted to the group, otherwise it makes the first entry into a new group.

A special condition is placed on multiple axles in the first group at the front of the vehicle. It will be split into two groups if either the rear (next) group is not a tandem (count =2) or if it is a tandem and the rear tandem separation doesn’t conform to the front tandem separation within the group delta. The explanation assumes the front group would be a tandem under these circumstances. If it is not, e.g. a tridem, it would also be split if the distance between the front and rear groups does not conform to the front group average, a condition that would occur in all likelihood. The first group, assumed to have a count of two, is then split into two single axle groups.

An alternate way of stating this special condition involves the designations A and B for the front axle group. The front axle group is A if it contains but one axle and B if it contains two. Assuming no C tractors, the conditions require a B tractor to have a rear tandem that conforms to the front tandem. A tractor that would be B otherwise becomes an A1 tractor if the front and rear tandems do not conform. The purpose of this rule is to allow two axle tractors to have a separation less than the group delta.

An alternate untested way of handling the special condition could be to have a group limit for the first group that is smaller than the group limit at large.

The group extent of any group is the length from the center of the group to the previous group center. The group length of the first group is zero.

Unit definition

Units are determined by applying the following rules in order until all units are designated. For each unit, the first unit designation takes priority over any later rule that also applies.

The tractor unit is made out of the first two groups. The next two groups are tentatively considered to be the tongue and body groups.

At the end of each unit, if only one group is left, the designation is a semi-trailer.

At the end of each unit, if the assumed body group extent is at least as long as the assumed tongue group extent, then the designation is a full trailer unit consisting of the two groups: tongue and body. Note that the meaning of the body group extent depends on whether the body group is or is not the last group.

At the end of each unit, if two groups are left and the body length is longer than that (DOLMAX= 11.9’) needed for a fifth-wheel dolly or spread tandems then the designation is a full trailer.

At the end of each unit, if tongue and body groups each have one axle then the designation is a semi-trailer.

At the end of each unit,
the designation is a one-group semi-trailer.

Making the structure code

Work through the units of the vehicles from front to back. Here NOA is the number of axles in the group.

```
alpha-numeric code
  tractor:
    first axle group
      one axle:    A
      two axles:   B
      three axles: C
      etc.
    subsequent groups: NOA
  trailer
    first axle group
      full
        one axle:     A
        two axles:    B
        three axles:  C
        etc.
      semi:           *NOA
    subsequent groups: NOA
```

Prospective Hybrid Classification Scheme

We propose to evaluate the effects of a classification scheme that takes the best parts of the modified E1572 standard coupled with the bin by bin editing of the probabilistic neural net(PNN).

The E1752 standard as augmented in our method above provides nearly 90% accuracy for all vehicles with more than two axles. The two axle vehicles are especially prone to having regions of overlap with two axle vehicles in other classes. Because of the overlap problem with the two-axle vehicles, bin editing as described here is proposed as an addition that will remove bias in the predictions.

More accurate results are highly desirable for understanding underlying phenomena of traffic flow. If the underlying features are related to imprecisions in the reported predictions, the next level of understanding cannot be achieved. Therefore any means possible for lowering data errors is welcome.

The bin procedure is to split the wheel base range of two axle vehicles into one hundred 10-cm bins varying from 0 to 10 m. We then count the number of each of the first five classes in each bin for a training set of data having ground truth. Based on those numbers, an estimate of the probability of each class is made for each bin.

The counting procedure has two manifestations: (1) summary and (2) raw. The raw method of population estimation is for a traffic counting device to report the number of vehicles in Classes 6 to 13 and the number of vehicles in each of the hundred bins.

The summary method requires a known calibration for the site which gives the class probabilities for each bin. These class probabilities can be added to the class total for each of the classes having vehicles in the calibration bin. For an axle separation falling in one of the overlap bins, fractions of vehicles would be added to two or more classes, the total of the fractions summing to unity. These results would be rounded to the nearest integer for reporting purposes.

Inherent in utilizing the methodology is a commitment to developing an understanding of the critical aspects of the calibration procedure.

We will begin that process by looking at two calibration methods for the Georgia data set.
Method I training and testing gives at most 10-15% errors on Classes 4 and 8 and does better on every other class. Method II training and testing does considerably worse on all data sets, with errors of -50% of Class 4 for the 9/11 data. Method I training and Method II testing is better than pure Method II: errors on Class 4 on the previously worse data set are -30%.

Results of this method are not given here because we obtain results with the *improved fuzzy method* described next.

Extracted Neural-Network Articles

The following articles have been extracted from progress reports from issues of Traffic Monitoring Progress (TMP) 6/95 to 12/96. They serve as a complete reference to TMP articles on the subject of the Neural Networks, giving details of items mentioned in the In-Depth Review of Neural Networks. The articles appear in chronological order. Minor editing has occurred including renumbering figures and tables, providing uniform titles, and clarification of context.

8/95

Nearest Neighbor Classification Applied to Georgia Data for 4 Axles

The intrinsic overlap and sensitivity of a set of data can be described by a nearest neighbor classification technique. This technique has been applied to the Georgia Tech data collection of May and September 1993.

The nearest neighbor method for the 4-axle data may be thought of in terms of points in the data space. The separations between the four axles are the three intervals. These three axle spacings may be considered as a point in a three dimensional space. The x, y. and z axes of the space correspond to the three axle spacings.

First, a distance is defined for pairs of points. Next a point is taken as a test point and is classified using the complete data set with that test point removed. The nearest point of the remaining data is then selected and used as the nearest neighbor identity of the test point.

Each point in the original data set is identified by its nearest neighbor point. If points for a particular class lie together, they will tend to be nearest neighbors. If two groups of points commingle, such as Class 3 and Class 5, they will tend to be nearest neighbors of each other in a nearly symmetric fashion.

Meaning of Off-Diagonal Elements

Observe in Table 1 that wherever the off-diagonal element (lightly shaded region) is greater than 5, the symmetrically placed combination is greater than 5. Thus, A Class 8 vehicle appears as nearest neighbor to Class 3 a total 7 times and a Class 3 appears as nearest neighbor to a Class 8 a total of 8 times.

What these overlaps suggest is that there is an intrinsic mix-up of the classes with off-diagonal elements. For the case of Class 8 and Class 3, all overlaps are less than 15% of either class. Bear in mind that this case is for 4-axle vehicles so that Class 3 is typically a pickup towing a car or light trailer. In Class 3, the fraction of such vehicles is small to begin with, so that the overlap with Class 8 and Class 5 occurring for 4 axle vehicles is quite negligible.

Nearest Neighbor Goodness

The results of the nearest neighbor classification can be used as a measure of the intrinsic overlap of the data. Any other classification technique, using the same three pairs of axle separations, will tend to have an occurrence matrix with off-diagonal elements at least as large as or larger than these. A Classification technique with off-diagonal matrix elements substantially larger than the nearest neighbor technique needs improvement.

Indeed, the axle separations that the vendor is trying to classify can be tested for their nearest neighbor in the standard database as a double check on his work.

Computational Difficulties

Although the nearest neighbor technique is a powerful method of classifying unknown vehicles, it is computationally inefficient. Each classification involves comparing each vehicle to be identified to every other vehicle in the database. The more complete the database of neighbors, the more time-consuming the computation.

Neural Networks Speed Up the Classification Process

The connectionist hypercube neural network used in this project is geared to make the nearest neighbor computation more efficient. The hypercubes are regions surrounding data points that represent the influence domain of those points for classification purposes. Any data point to be classified that falls within the influence domain of a database point is considered the nearest neighbor of that database point. Influence domains of nearby database points of the same classification are merged together for classification purposes.

The merge process speeds the computation but loses the ability to single out that vehicle in the database that is closest to the one being identified.

Added benefit

The nearly symmetric property of the nearest neighbor method of classification has an added benefit. For applications looking at exposures or overall error minimization for a large collection of data, cancellation of errors can be used to advantage. A utility analysis shows that the errors incurred by off-diagonal elements cancel if the occurrence matrix is purely symmetric. This added benefit of the nearest neighbor approach makes this technique highly desirable for these purposes.

Need for an On-line Data Base

Manufacturers who build classification equipment need a means to check their equipment’s ability to classify vehicles. An Internet on-line data base would be able to provide a manufacturer a means of checking the nearest neighbor classification of a vehicle. The Internet page for this service could start as a form that is filled out with axle spacings. The computation of the nearest neighbor occurs by a so-called *cgi* file that returns the nearest neighbor identification of the vehicle.

The usefulness of the system depends directly on how reliable the data base is. Criteria for acceptance of data into the data base can be established. Editing checks on the data can be standard fare.

12/95

Symmetry Of Nearest Neighbor Performance Matrix

In the August 1995 progress report it was stated that the nearest neighbor technique yielded symmetric matrix elements for the performance matrix. The utility analysis referred to there suggests that symmetric matrices are highly valuable for classification purposes. The close relationship of the nearest neighbor technique to our neural network method suggests that the latter also has nearly symmetric matrices. Having recently computed highly unsymmetrical nearest neighbor performance matrices under, what turned out to be, very specialized conditions, we have investigated the conditions under which the symmetry holds. Sufficient conditions exist that are quite simple and hold up under a variety of common circumstances. First for didactic simplicity, let us think of explaining classification for two-axle vehicles in light of the four-axle computations. One condition, which the four axle vehicles from the Georgia data set satisfied, was that there was no apparent ordering of the data with respect to class. The algorithm was written so that if two vehicles showed the same difference in axle spacing, the last one was selected. Now, because the last vehicle is not more likely to be of one class or another, or more likely to be at the end of the data set than elsewhere, no bias is introduced by data order.

Had we sorted the four-axle data with all the Class 2 vehicles before the Class 3 vehicles, there would have been a bias toward Class 3.

To avoid this type of problem, we have modified the nearest neighbor algorithm to give fractional assignments if more than one vehicle has the minimum distance in axle spacing. In this case, if there are ten vehicles having the same minimum distance to the test axle spacing, the class of each one of the ten is given a partial weight of 1/10. Thus, if there were to be eight Class 2 vehicles and two Class 3 vehicles at exactly the same distance of spacing, an increment of 0.8 of a vehicle would be given to Class 2 and 0.2 of a vehicle to Class 3. The above effect has an algorithmic nature. There are other conditions not associated with our algorithm that affect the symmetry of the performance matrix. These conditions are required when there are models of vehicles that have more than one representative with the same spacing. Let us give an example of this type of effect.

Suppose that in a hypothetical country there are only a few pickup trucks having axle spacings that overlap with passenger-car axle spacings. Furthermore, suppose that the passenger cars come in only ten varieties with two or three cars in each model. Then with a handful of pickups, none of which have a spacing exactly the same as those of the models, the performance matrix would be highly skewed. Let us explain.

In our hypothetical case, all 25 of the Class 2 vehicles would have other Class 2 vehicles as their nearest neighbor. The Class 3 truck in the overlap region would most likely have a Class 2 vehicle as its nearest neighbor. Suppose there were 10 pickups in the overlap region, each being distinct models and 20 in the pickup-only portion of space. Then the performance matrix would be very much like that shown in Table 1, where the minus sign indicates that the number might be a few less pickups and the + sign indicates possibly a few more. (It might be that only 5 of the 10 trucks in the overlap region had cars as nearest neighbors, then the Class 3 row would read 5 and 25.)

Table 1: Hypothetical Skewed Performance Matrix

given\guess	Class 2	Class 3
Class 2	25	0
Class 3	10-	20+

This matrix says that all the Class 2 vehicles are nearest to Class 2, but a substantial number of the Class 3 vehicles are also nearest to Class 2. Therefore, the shaded regions in the matrix have numbers that are unequal and the matrix is unsymmetrical in the technical sense.

Now, we can state the sufficient condition for a symmetric performance matrix. Suppose that in any small but finite axle-spacing interval (bin) there are N1 vehicles of one class and N2 vehicles of a second class. Furthermore, suppose that each of the vehicles in the various classes i.e. equally likely to occur anywhere within the interval. Then if follows that the performance matrix will be symmetric. We will demonstrate this below.

Furthermore, suppose that any continuous distribution of vehicles can be divided into subintervals, each of which meets our previous sufficient condition. Then, it follows by superposition, that the sum over all intervals will also result in a symmetric matrix.

The symmetric condition can be shown as follows: Consider a vehicle in the first class. Then there are N1-1 vehicles of the class and N2 vehicles of the other class that might be nearest neighbors. If there is an arrangement where a particular vehicle of a particular class is the nearest neighbor, that vehicle has a chance of 1 out of N1-1+N2 vehicles of being a particular vehicle. That is, the chance of the nearest neighbor being of the first class is (N1-1)/(N1-1+N2). The chance of it being the second class is likewise N2/(N1-1+N2). Thus of all N1 vehicles in the first class, we would expect N1N2/(N1-1+N2) to be identified as in the second class.

A similar argument shows that N2N1/(N2-1+N1) vehicles on the second class are identified as being in the first class. These two numbers are the same. The performance matrix, thus, looks like that in Table 2:

Table 2: Symmetric performance matrix for the sufficient conditions stated above

given\guess	guess class 1	guess class 2
given class 1	$N_1(N_1\text{-}1)/(N_1+N_2\text{-}1)$	$N_1N_2/(N_1+N_2\text{-}1)$
given class 2	$N_1N_2/(N_1+N_2\text{-}1)$	$N_2(N_1\text{-}1)/(N_1+N_2\text{-}1)$

When N1 and N2 are large, the off-diagonal matrix element is the geometric mean of the other two numbers.

Thus, the skewed performance matrix is indicative of a non-uniform distribution of vehicles over a given interval in the nearest neighbor method. In the skewed example given, the presence of a Class 2 vehicle indicated that it was likely that another Class 2 vehicle of the same model and axle spacing would also be present.

Whether or not highly skewed performance matrices occur in practice can be taken as an indication of whether or not there is sufficient measurement precision of the axle spacing to resolve distinct models. The actual axle spectrum can be generated by using manufacture’s data along with populations for each model. Such a calculation would lead to the requirements for measurement precision to obtain model resolution.

If the measurement error of axle spacing is large, then that error distributes the vehicle nearly equally over a substantial portion of the error interval. If the error interval is large compared to the separation between models, then it is as if vehicles were spread uniformly through intervals and the symmetry condition would be satisfied.

Treatment of the model limit

Here we show that in the limit where the number of vehicles is much larger than the number of classes and one class occurs in models with discrete spacings and the other class in models that have equally likely spacing over the entire interval, a large asymmetry is induced in the performance matrix. If however the number of classes is large compared to the number of vehicles, a symmetric performance matrix results.

Suppose we have an interval [0,1] with two classes of objects at various locations. One of the classes is randomly distributed uniformly throughout the interval. The second class of objects is made up of objects in various model categories. These models are distributed at random on the interval, but there are a finite number of models M. For this particular calculation, we assume that all M models are equally likely. Further, we assume that there are N vehicles in each of the two classes. Then we ask, “What is the nearest neighbor performance matrix for the system?”

There are tow limiting cases: (i) When N >> M, then there are many (N/M) vehicles of each model for the second class. Thus, all N of the second class vehicles will have a second class vehicle as a nearest neighbor, and none will have the first class as a nearest neighbor. Under these conditions, the second class vehicles appear as M distinct vehicles to a first class vehicle. Thus, the first class model in this limit is equivalent to a model where M second class vehicles are distributed uniformly throughout the interval. The performance matrix in this case is given in Table 3:

Table 3: Model limit: N>>M

	first class guess	second class guess
first class given	M(M-1)/(N+M-1)	MN/(N+M-1)
second class given	0	N

(ii) The second limiting case is when M>>N. Then the fact that second class vehicles appear in the form of models is not apparent. In this case, there is a full symmetry of the matrix which is independent of M as shown in Table 4:

Table 4: Model Limit: M>>N

	first class guess	second class guess
first class given	N(N-1)/(2N-1)	N^2/(2N-1)
second class given	N^2/(2N-1)	N(N-1)/(2N-1)

The general case

A general case could be formulated and treated analytically by similar methods, but this case is beyond the scope of the current project. It would serve to fully quantify the nature of the asymmetry.

‘

Spring/96

How A Neural Network Works Geometrically

A feed-forward neural network operates on the basis of some very simple principles that are already familiar to many of us. The usual description is in terms of subset of mathematics. To lessen the dependence on mathematics, we give a presentation of a feed-forward neural network in geometric terms. Unfortunately, a lot of attention is required for understanding them completely. We summarize the conclusions here. We find we can construct a general two-dimensional function using neural networks that are constructed with neurons that have two inputs, one output, and either fire or do not. These functions can be represented in black and white graphics on plain paper produced, for instance, by a Fax machine.

The standard feed-forward binary neural network with hard thresholds is a very powerful computing element. A neural network is, in general, what is referred to as a universal finite computer. That means a neural network can be designed that does the same logic that any computer can do. The simplest of such computers is a Turing machine and a more complicated one is a modern Cray computer. Those computers are a very special case of feed forward neural networks involving black and white logic augmented by recurrence or feedback.

To understand what a hard threshold is, we examine a very simple neural network. This consists of a cell body having a neuron modeled to be the node with the axons as inputs and the dendrite composites as outputs.

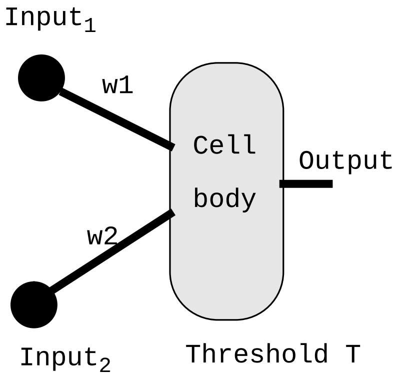


Fig. 1 A neural network node with a hard threshold

The neuron model shown has two inputs, and one output, a condition that simplifies the description but still keeps all the interesting neural network behavior. The two inputs values are numbers such as 1.43 or 76. The first input is multiplied by the first weight and is summed with the second input multiplied by its weight. Thus in the example if both the weights are unity, the input sum is 77.43.

If this sum is larger than or equal to the threshold T, the output of the cell body to dendrite composite is represented by the output line of Fig. 1 and has the value of unity. If the weighted input sum is below threshold, we say the neuron doesn’t fire and the output is 0.

If the threshold is not hard, by definition the output of the neuron takes on a range of values that depends on the threshold and other parameters. Rather than having black and white output, it is grey-scale.

The geometric description of the neural network is based on the following observations: (1) Input1 and Input2 can be regarded as the x and y coordinates of a Cartesian grid. Such a grid can be on a piece of paper with its origin at the center of the paper. A single pair of x and y points correspond to a point on the page. (2) The output is zero or unity, which we can represent as either a lack of a blackened pixel or the presence of one for each point on the paper. Thus the output of any neural network with two inputs can be thought of as a black and white picture such as might be sent by Fax. We have yet to establish that any such Fax can be produced. This neural network can be made up of lots of neural network nodes of the variety of the one shown in Fig.1. They are hooked together in tinker-toy fashion with the output plugging into the input of another unit.

First consider the case of a single node with w1 =1, w2 =0 and T=1. Only the x value matters in determining the output because the y value is multiplied by zero. For the product of x and w1 to be greater than or equal T, x must be greater than or equal to T. If we color this region textured black, we obtain what we see in Fig.2.

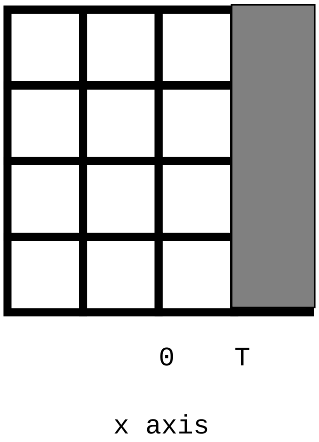


Fig. 2 The region x >= T.

On the other hand, if we set w1 =0 and w2 =1, only the y input matters and we obtain a graph similar to Fig. 2 except that it is rotated counterclockwise 90 degrees as in Fig. 3.

If both w1 and w2 are set with magnitudes of unity, the region shaded has a boundary at plus or minus 45^0. One case is shown in the diagram of Fig. 4.

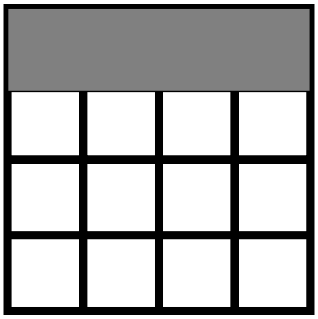


Fig. 3 The region for w1 =0 and w2 =1, T =1.

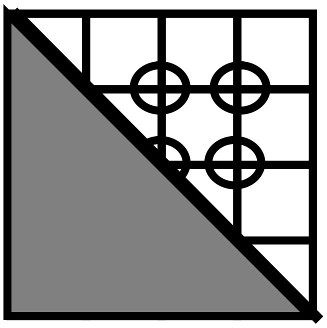


Fig.4 The region for w1 = 1 and w2 = 1, T=0. The four circled points are for x,y each of (0,1).

The pattern that emerges is that a single neuron can compose a region that darkens any half plane. A small technical point is that when darkening an upper half plane, the line y = T is darkened as in Fig. 3. To darken a lower half plane, a negative weight w2 = -1 is used with a positive threshold. Again the edge of the plane that is darkened, y= -T, is also included in the darkened region. Likewise in Fig.4 the origin is included in the boundary. In this case increasing the threshold will increase the value of x+y of the boundary. A threshold of 1/2 would give an intercept of both the x and y axis at 1/2. That is, x=1/2, y=0 meets the firing condition of a threshold of 1/2 as does y=1/2, x=0. When the boundary is part of the region, the boundary is said to be part of a locally closed region.

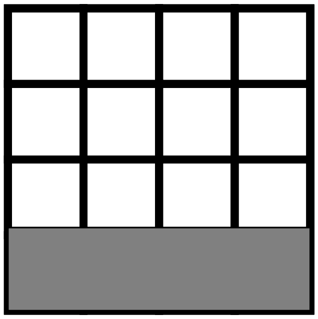


Fig. 5 The region for w1 = 0, w2 = -1, T=1.

The upper boundary in the lower half plane of Fig.5 is also locally closed because the neuron fires for y = -1, giving y*w2 =1, thereby meeting the firing condition.

Changing the threshold in any of the figures drawn so far will translate the textured black region drawn. Thus, a higher T in Fig. 3 will have a shaded region higher than that drawn. We see that the threshold parameter controls the movement of the shaded portion of the drawing to a new location with the new edge parallel to the old.

With w1 not equal to w2, the boundary is skewed at an angle that can be determined by analytical means. We have seen boundaries that are horizontal, vertical, and at 45^0. Boundaries at any angle can be generated by so adjusting w1 and w2.

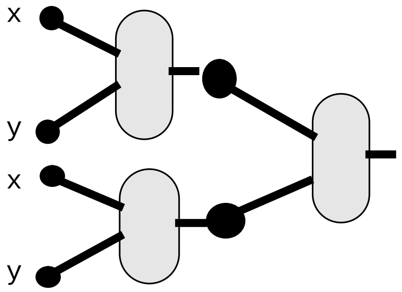


Fig. 6 A double layer of neurons with 2 in the first layer and 1 in the second layer.

Double Layer of Neural Bodies

Next we consider a double layer of neurons as shown in Fig. 6. This neuron set is special in that each pair of inputs goes to only one neuron. In the general case, the inputs are allowed to go to any of the neural cell bodies in the appropriate layer. In that case four inputs would be compared against the threshold rather than two inputs. The general case would have to be interpreted in four dimensional space, a task quite difficult to visualize. With two pairs of nodes, having the two x values connected and the two y values connected, we describe a two dimensional surface, in particular, the plane of the paper. We may draw the connected neural net as shown in Fig.7.

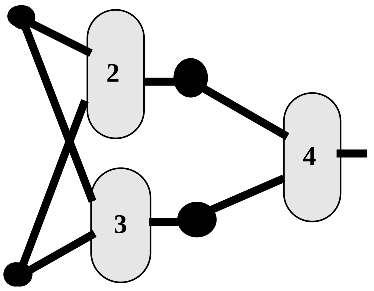


Fig. 7 A redrawn Fig.6.

Here, the two inputs are connected to each of the neural bodies in the first layer. In layer 1, we can take the upper neuron as being that which generates Fig.2. The lower neuron we will take to be the one for Fig. 3. The one in layer 2 we will take to be the one in Fig. 4.

Here we determine the bitmap (Fax) diagram for Fig. 7. The two neurons in layer 1 have outputs of zero or unity. The neuron in layer 2 takes those outputs and combines them according to the rule of the second layer neuron. Unlike the first layer, there are now only four possible pairs on inputs: (0,0), (0,1), (1,0) and (1,1). The possible outputs for the second layer are shown in Fig. 4 as the circled points. Only those four points are used in determining the map. The outputs of the second layer of neurons are always 0 or 1.

The rest of the map in Fig. 4 is useful because it tells what would happen if the layer 2 neuron parameters were changed. We have already noted that if the threshold is changed to 1/2 for this neuron, the darkened pattern moves north-east until it intercepts the x and y axes at a value of 1/2. If the value is increased to 3/2 for the threshold, all points except that for x=1 and y=1 would be in the dark region.

Let's determine the pattern for the threshold level of 1/2 (the same output as for threshold of 0 shown in Fig.4). If no neurons fire into layer 2, we are at point (0,0), a black point inside the boundary of the black region. The no-neuron firing regions are the white regions in Figs. 2 and 3. Thus if both Fig. 2 and Fig. 3 are white at the same x and y location, the input will be zero and the figure generated will be black at that point.

Conversely, if one of the figures has a black point and another a white point, we will be at (0,1) or (1,0) which is in the white region of Fig. 3. Thus, wherever there is a black region, in Figs. 2 or 3, a white region is generated. The resulting diagram is shown below in Fig. 8. It is the intersection of two half planes.

The operation of combining the two figures can be stated as INVERTED Fig. 2 AND INVERTED Fig. 3. By the term INVERTED, we mean that the black becomes white and the white becomes black. OR means that if either of the inverted regions is black, both are black. That is, black OR white is black, and white OR black is black, although not used above. The other term is AND. In combining two regions, white AND black is white. Also black AND white is white, whereas black AND black is black and white AND white is white.

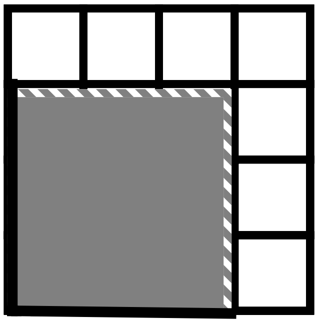


Fig. 8 The result of Fig. 7 with node assignments of Figs. 2, 3 and 4. The darkened region should extend up to but not including the solid lines of the interior. If the drawing area were large enough, the black region in the SouthWest direction would extend to infinity. We call the region an open quadrant.

The reader by now should have an inkling of the power of neural nets to produce a variety of behavior. We have shown that by using Fig. 4 in layer 2, we can combine any two drawings such as Figs. 2 and 3 by inverting them and using the AND operator. Lots of other combinations are possible. If we could replace Fig. 4 by one in which only the point (1,1) is darkened, then only if both points in the input figures are black will the output be black. This simply would be the AND operation without inversion.
For a complete understanding of the subject, the reader should construct values of the weights and threshold that produce such a neural node. He will also be interested in using Fig. 2 or 3 in the second layer. The reader who simply wants an overall understanding of what it is that is claimed for neural nets, may proceed on faith.

The Fig. 3 illustration suggests that a given neuron can have one of eight possible outcomes. Four of these are to have 3 of the four cases (each indicated by a circle) white and the other is to have three of the four cases to be black. For a finite sheet of paper, it should be evident that those operations are redundant.

AND and OR operations

We suggest that restricting our attention to the AND and OR commands is sufficient for generation of any Fax in a multiple layered neural net. The AND command has the (1,1) location black and the others white. The OR command has the (0,0) location white and the others black. That these commands are sufficient follows from the fact that one can generate a black pixel at any location using the AND of two translated quadrants.

The quadrants are formed by the AND of two half planes. Thus, Fig. 9 can be a pixel generator. This same method can be used to generate a rectangle of arbitrary number of pixels.

Simple arguments also show that either closed boundaries or open boundaries can be produced.

By allowing only one INVERT operation just before the output of the net, figures with open black boundaries such as that in Fig. 8 can be constructed everywhere. To have multiple regions with different boundary properties, some of these can be generated by INVERT operations with their placement at earlier levels in the net.

The representation of a Fax

The paradigm should now be evident. Each of the nodes or bodies of the types shown in Fig. 1 are represented by a bitmap such as those shown in Figs. 2,3,4 and 5. We can construct any of the possible darkened half-plane regions by translating or rotating any of the already constructed regions by changing the weights and threshold. In each of these regions, the edge of the region, the half-plane edge, is considered to be part of the region.

The question that remains is: what combinations result from connections of nodes, the simplest of which is the two layer system shown in Fig.7. What is the class of Figures that result from the three layer system shown in Fig. 9 or one with an arbitrary number of layers?

A short digression on the architecture of these kinds of neurons is necessary. The two inputs in Fig. 9 fan out into 8 inputs for the nets and these require 4 nodes in the first layer. Without any fan out in the second layer output, the net must, of necessity, have two neurons for the four outputs. With fanout, the third layer can contain any number of neurons greater than one.

If we are dealing with N outputs from a given layer, the next layer must, of necessity, have at least N/2 outputs. Thus, a layer of 1024 outputs would require at least 10 layers to reduce it to a single binary output using information from all the nodes. With neural nets that allow an arbitrary number of inputs, an output binary signal can be obtained in one layer by connecting everything to the output neuron.

Now that we see the possible connections, we ask the simple question: can an arbitrary Fax be produced by half-plane regions, AND operations, OR operations and INVERSION. If so, it could be produced by a simple binary neural net. We can see that this is possible by constructing a single black rectangular region representing a pixel, for instance, using four AND operations. This pixel can be placed anywhere on the page. By combining many such pixels by the OR operation, an arbitrary Fax is produced.

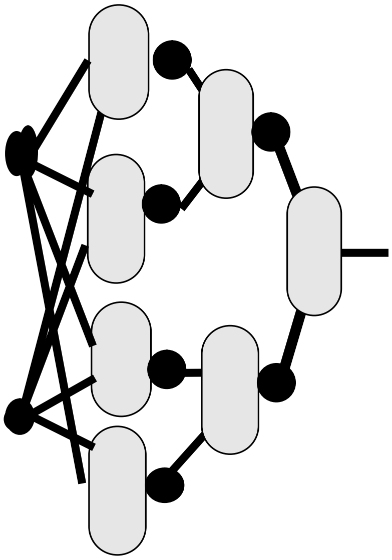


Fig. 9 A bitmap generator may give a finite Fax with two inputs and 4 nodes in layer 1, 2 nodes in the second layer and 1 output node.

Constructing the PNN

The Probabilistic Neural Network (PNN) has evolved from a number of stages of thinking which we will summarize here. The original concept for our classification problem started as a hypercube neural network. Because this neural network suffered from the need for individual judgements with respect to pruning of individual networks tuned to special sub-networks and lacked the fundamental concept of probability-based selection, a better way was sought.

The hypercube neural network had proceeded in a rather straight-forward manner. Special active nodes were selected at random for the purpose of representing a region considered to belong to a given class. These nodes would grow or shrink depending upon the location of neighboring points/regions of other classes. The technique worked very well when there was no overlap of the distributions of the classes.

But when distributions did overlap, the method produced unbelievable alternating regions side-by-side. A two class overlapped region could be thought of as being represented by a conglomeration of small rectangular black or white regions, which looked like a nearly homogeneous mixture of salt and pepper in amounts, hopefully in proportion to the local probability density of the respective classes. No guarantees were available that the local probability density was indeed approximated, although we demonstrated that this was the case under special circumstances involving an ensemble of such networks.

At best, such a network would assign a pure vehicle identification at a given point in the parameter space, rather than the probability of such an identification. The procedure would be valid if the experimental errors of the measurements were sufficiently small. In that case, it might be truly possible to tell if a car were of a particular make and model just by using the measured value. The measured axle-separation values have, however, substantial experimental errors (on the order of 10-cm at very best) that make such a determination currently impossible. Because many vehicles can fall at the same point in axle-separation space, one cannot say for sure what the true identity or class is. Thus, we are motivated to use a method that describes the probability of identification at each point of parameter space.

For a sufficiently large number of test points, if one took the average of many constructed hypercube neural networks using different random numbers to generate them from the same training data, they would (on average) predict the right proportions for the training data. That theorem gave no guarantee, however, that any one such neural network would do so. To enhance the possibilities that plots of the distribution of rectangles or hypercubes did appear correct to the eye, various choices of parameters existed for the networks. These choices led to individual attention required for each such network.

The idea of a PNN was attractive because it reduced the amount work required and provided a direct way of describing the truth of the data. The details could be incorporated in a number of ways, the most common of which was by using radial basis functions. At this time in the project we undertook two risks, the successful completion of which turned out to be crucial to the success of the PNN. The first was to submit a paper of our work on Super Synchronization to a Neural Network conference held at George Washington University, going beyond the scope of our agreed-to work. The second was to upgrade our programming skills from C programming to C++. How we did the work was not specified in our agreement.

At the neural network conference we became acquainted with a hash-based neural network that was said to be nearest-neighbor based. The details of the network were sketchy but it was clear it revolved around a hash-table. We generalized this description to that of a probabilistic neural network. Having just encountered hash tables in our C++ course, we were all set to proceed.

Another useful aspect of the work reported at the conference by Azim Eskandarian was using equally spaced bins for each of the independent variables. We generalized this by using fuzzy bins for more accuracy. For instance, if a point fell on the boundary of two bins, in our approach it was split equally between the bins, whereas in a pure bin approach it would have to be put in one bin or the other. In two-dimensional space this technique is known as *area-weighting*.

Combining these techniques led to our PNN. This PNN suffered, however, from one small drawback. Not all of the regions in space of the training data contained the test data. Using fuzzy bins decreased the number of unclassified points compared to hard bins because a larger portion of parameter space was involved by the fuzzy fringe, but there was still a small fraction of a percent of all the data that remained unclassified.

To handle the unclassified data, a second PNN was constructed explicitly for those points. This re-classification PNN had much coarser bins which when applied to the test data classified all but one point out of approximately 20,000. This was judged to be adequate.

Later work (reported in the June and July 1996 progress reports) looked for a way to gain acceptance of the results of the PNN through the mechanism of understanding the meaning of the neural network results. This idea was accomplished by using the results of the PNN to establish goals for the other more-understandable methods. Those methods were changed until their performance approached that of the PNN method.

6/96

If/Then Analysis of Georgia Data Set

Simplification of the Neural Network rules presented at the NATDAC ‘96 conference held in Albuquerque in May can make them more amenable to general understanding and ultimate acceptance. The if/then analysis we are now undertaking is one of many possible forms.

Strictly speaking, *any* computer algorithm falls within the if/then paradigm because it is based on logic. The amount and complexity of the logic in a computer program is generally much greater than we are referring to here.

A field of study known as induction analysis has developed the goal of automatically deriving a set of human-understandable rules from a set of data. This field has been directed at both discreet and continuous data. The 13 FHWA classes are discrete and the axle separations are continuous. Typically, the approach is to utilize a set of hierarchical binary decisions to arrive at a classification. These techniques are known as decision trees.

Decision trees come in two varieties: axis parallel and oblique. The axis parallel approach makes decisions based on being above or below single values of continuous variables. An example is an axle separation of 304.8-cm (10.00’) used in the Scheme F classification developed by the Maine Transportation Department.

In two regions in the space of axle separations, such a condition can be represented as a plane perpendicular to the first axle separation axis. This plane is parallel to the axes except the one to which it is perpendicular. If a similar plane is used to divide space, but the plane is rotated so that it is not parallel or perpendicular to at least one axis, it is known as an oblique separation.

An example of an oblique condition is that the sum of axle separations be less or greater than a given amount. We would like to have a method that could find such a condition.

The OC1 Decision Tree

We have obtained and installed an oblique algorithm known as OC1 on our current IPC Sun Computer. This algorithm is described in the AAAI-93 paper by Sreerama K. Murthy, Simon Kasif, Steven Salzberg and Richard Beigel entitled ***A System For Induction of Oblique Decision Trees.*** The algorithm was written in ANSI C while the authors were at John Hopkins University.

The algorithm has three stages, the first, the axis parallel separation, it does at each level. In the second stage, the algorithm perturbs the angles of the separation plane with a hill-climbing algorithm and takes small random jumps in the location parameters to find a better division of the remaining data. Having built the decision tree, the last stage is called pruning. Pruning prevents the space being divided into regions having a single data point. It combats the problem of overfitting a particular data set. A data set can be diagnosed as overfit, when the predictions of a new data set of similar data are not accurate.

The algorithm has a number of options. Axis parallel operation without reorientation of the planes is one of them. Various ways to divide up the space are also available. Most of these methods are based on reducing the impurity in each of the classification regions. Well known methods such as the Gini index, the Max Majority measure, the Sum Minority, the Quinlan Information Gain method, the sum of Variances, and the Breiman Twoing Rule are used to define the impurity or its inverse.

OC1 Pruning Problem

The pruning method utilized by OC1 is a Breiman Cost-Complexity technique that minimizes the sum of (a cost for) misclassifications on a test set of data summed with a fractional multiple of complexity (the number of leaves required in the decision tree). The particular pruning method chosen by the authors of OC1 proved unsuccessful on our full Georgia data set. This apparently has to do with the details of the pruning method which doesn't work when there is substantial replicate data. Because the replicate data occurs primarily on the two-axle vehicles, we were able to run and test the algorithm on data with greater than two axles.

Although, the software doesn't compute a performance matrix that would allow comparison with a matrix computed by our probabilistic neural network reported at NATDAC, we are able to compare the total misclassifications. For the first test set of two axle data, OC1 (Twoing Rule and 20% pruning) resulted in the following misclassifications:

Class 0:	(3/3)
Class 2:	(15/16)
Class 3;	(33/229)
Class 4;	(4/18)
Class 5:	(20/38)

Class 6:	18/231	:18.5
Class 7:	3/3	:1.0
Class 8:	7/214	:23.4
Class 9:	3/2263	:3.7
Class 10:	7/31	:0.7
Class 11:	0/66	:1.0
Class 12:	6/19	:0.2
Class 13:	0/0	:0.0

Those vehicles appearing in Classes 2,3, and 5 are two axle vehicles pulling trailers. The class 4 vehicles are 3 axle buses. The values for these classes are put in parentheses because they leave out the two axle cases.

A comparison is given with Class 6 and above on the same data using the bins reported at NATDAC using the neural network described there with reclassification. The numbers following the second colon on a line should be compared to number preceding the /. We can see that the reclassification neural network does considerably better on Classes 10 and 12 but does worse on Class 8.

Are Oblique Planes Understandable?

The coefficients of the various axle spacings using the OC1 method described above involve non-integer values. Furthermore, the number of oblique hyperplanes produced by the method is 11 out of a total of 15. Each of these 11 used coefficients for either 5 or 6 axle values.

Just how understandable are these results?

The meaning of four single dimension values obtained is within the context of regions already determined by the oblique hyperplanes. Within those regions of parameter space, the simple rules performed better than any rotation of the hyperplane.

An example is a rule for the second axle separation being above or below the value of 11.95. A Value above separates out a Class 11 -only part of space. Here 47 of the 66 Class 11 vehicles reside. That part of space is defined by two oblique conditions:
$-0.05 x[1] + 0.63 x[2] -0.08 x[3] -0.50 x[4] + 0.57 x[5] + 0.21 x[6] + 0.588996 > 0$

and

$0.16 x[1] -0.17 x[2] - 0.07 x[3] + 0.41 x[4] + 0.15 x[5] + 0.34 x[6] - 0.344741 > 0$ and one non-oblique condition:
$x[4] > 1.695$.

Here x[1] indicates the first axle separation, x[2], the second, etc.

The first oblique condition serves to separate out Class 9 vehicles in a separate region. The second oblique condition relates to differentiating between Class 6 and Class 8 vehicles. Neither of these oblique conditions involves the simple concept of the vehicle length. The non-oblique condition encountered means that the x[4] value is above zero, i.e. there are at least 4 axle separations or 5 axles on the vehicle.

Thus, the rule for Class 11 loosely means that where there are few Class 9 vehicles and many Class 6 vehicles, and considering only vehicles with at least 5 axles, when the second axle separation is greater than 11.95 (30.48cm units) the vehicle is a Class 11.

We do not have a simple interpretation of the two oblique conditions. Because they are in six dimensional space, this task looks very difficult. We will use axis parallel techniques to help clarify the situation in July.

7/96

Decision Tree Classification Methodology

In June, oblique decision trees were applied to simplify the conceptual complexity of the probabilistic neural network. In that work, it was found that the decision tree concepts were, by necessity, complicated and the implied model was *non-sequitur*. For instance, the classification of Class 11 was apparently dependent on how Class 9, 6, and 8 vehicles are classified. The causal inference would have to support such a hierarchy to make it understandable. This would be the case if something in the spacing of the three axles of a Class 6, single unit truck, influenced the spacing of all five axles of a tractor/double trailer unit.

To be understandable, the mechanism for any underlying dependencies needs to be clarified. Such an explanation has to do with the manufacturing process of trucks. For instance, the presence of a set-back tractor axle increases fuel efficiency, payload capacity, maneuverability and allows for an integral sleeper compartment. The second axle of a tractor may have forward or axle-back positions. [http://www.volvotrx.com/catalog1.html]. Regulations also play a role in determining overall length, [http://www.ontruck.org, industry issues, weights & dimensions] for instance, the 14.6 m [48'] trailers known as *turnpike doubles* with vehicle lengths up to 34.7 m [114'].

In principal, an expert in tractor and trailer manufacturing could provide the rational for the length choices of bodies and support point offsets. Such a task would also be of benefit to the manufacturer in reviewing marketing supply possibilities. Demand experts could also be useful in such a study. Understanding the choices of lengths and the resultant implications for axle location and its variability could be helpful in anticipating changes to classification schemes. The overall benefit would be to increase the understandability of current and future classification schemes.

Parameter Estimation

Is there a substantially better method of analysis of axle separation data that can be used on a daily basis? A direct Monte Carlo model calculation suggests that there is. The method would require a fundamental shift in the data storage practice based on truth in data.

Unbiased estimates of traffic characteristics over various periods of time are useful for a variety of applications. These range from prediction of road repair and renewal, congestion anticipation, and real time traffic reports to exposure levels of unsafe vehicle combinations.

The length of observation time determines the sample size and, thus, the accuracy of the inferences. In assessing congestion, for instance, the number of vehicles in a fifteen minute sample is considerably smaller than that required for average daily traffic.

Analysis of traffic patterns now includes separating the traffic flow into various classes of vehicles over short periods of time. How short a period of time can be used with reasonable accuracy depends on the ability to classify accurately. The details of the classification scheme are intimately tied to the accuracy obtained. For instance, a simple if/then scheme may be easier to implement but be less accurate than a probabilistic neural network.

Here we report on the effect of various editing procedures, assuming perfect measurement of independent variables such as axle separations. The crux of the problem concerns the classification of vehicles falling in an overlap region that has no clear class distinction. The results of no editing, use of an edit matrix, and least square editing are compared within the context of a simple model.

Probability Model

The hypothetical probability distribution function model for each class is taken to be trapezoidal in shape, but specialized to triangular form for this study. The abscissa may be thought of as the axle separation of a two axle vehicle. Two triangles are chosen that have overlapping areas as shown in Fig. 10.

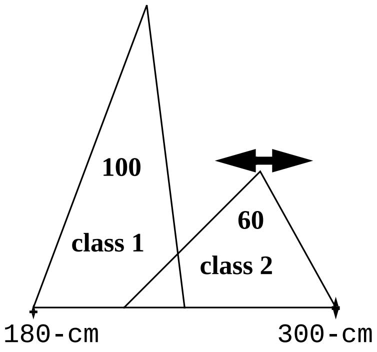


Fig. 10 Model A

Class 1 is fixed and will have random samples drawn of 100 vehicles. The shape of class 2 is also fixed and samples will be drawn of size 60. The distribution for class 2 has an uncertainty of its location with respect to class 1. In Model A, the distribution is fixed at the location shown. In Model B it is moved 40-cm to the right. The actual distribution of translations for class 2 may be thought of as a tophat distribution, uniformly distributed between plus or minus 40-cm. The different translations are site (or time) dependent, and the distribution of sites leads to the distribution of translations. At a given site there is a fixed distribution at perhaps a different displacement than that in Model B, however, here we will look at that site described by Model B.

This model approach has a number of aspects encountered in practical vehicle counting situations. First the distribution of vehicles with respect to axle separation can vary considerably across the country or with time of day (or longer time measure). This means that a counter-classifier tuned for one local and one time of day will not necessarily be tuned at another local or a different time of day.

First: Model A Only

To put the results in perspective and to introduce the edit matrix, we assume that class 2 is fixed as shown in the Model A location and that all other sites are also described by Model A. We then consider various vertical demarcation lines, (usually in the overlap region) a point on which could serve as the base, chosen from an interval beginning at the left side of the support of class 2 to the right side of the support of class 1.

After choosing a demarcation line, we count the number of vehicles having axle separations falling on either side of the line. These are the raw estimates of the class 1 and class 2 populations, rl and rr respectively.

We compute an edit matrix which tells what fraction of each count corresponds to class 1 and what fraction to class 2. This edit matrix is based on the Model A diagram which has an area under the curve for class 1 of 100 and for class 2 of 60. The area under the curve of class 1 and to the right of the demarcation line is a1r, while that for class 2 is a2r. Likewise the area to the left of the line is a1l for class 1 and a2l for class 2. The fraction of vehicles to the right of the line that belong to class 1 is then
f1r = a1r/(a1r+a2r) and that for class 2 is
f2r = a2r/(a1r+a2r). By replacing r by l we can use the same equations to give us f1l and f2l. We then

obtain an unbiased estimate of the population in class 1, N1 and that in class 2, N2:

N1 = rl*f1l + rr*f1r,
N2 = rl*f2l + rr*f2r.

The coefficients f1l, f1r, f2l, and f2r which combine with the [rl,rr] vector are the edit matrix. An application of the edit matrix could be to edit Scheme F data, which is acquired by a demarcation line system.

Of the two schemes described above, only the edit matrix method is unbiased. These biases can be computed by breaking up the axle separations into bins. For these calculations, we split the region between the left side of the support of class 2 and the right side of the support of class 1 into 5 equally spaced regions. The grid extends to cover the remainder of the supported region.

The left, top, and right side of the class 1 distribution have abscissa values of 180-cm, 225-cm, and 240-cm respectively. Likewise, those of class 2 are 216-cm, 270-cm, and 300-cm. We generate random events by choosing either class 1 or class 2 in the ratio of 100:60. Then we choose a bin at random with each distribution bin being chosen according to the area under the class curve for that bin. A weight is assigned to the bin for least square estimation purposes that is the inverse of an estimate of the population variance based on the model of an independent binomial distribution of bin populations.

At a demarcation line location of 201.6-cm, the raw score is in error of -83.1 +-6.6 for class 1 and negative that amount for class 2. By contrast, the edit matrix value is in error by -0.37 +- 6.2. These are obtained by averaging 1000 trials of 160 vehicles each.

Moving over in bin widths at a demarcation line location of 249.6-cm, the most accurate raw prediction error is
2.48 +- 2.94 compared to the edit matrix value of
0.13 +- 2.92. In between the two cases reported the error varies smoothly.

Two additional methods were also used. The edit matrix was applied bin by bin giving 0.22 +- 2.39 (independent of demarcation line). A weighted least squares technique gives
-2.7 +- 2.7.

The unbiased estimate with the smallest standard deviation is the best in that the standard deviation is a measure of how much additional data is required to lower the standard deviation to a given value. Thus, the unbiased methods are best to worse:
(1) bin by bin edit matrix
(2) weighted least squares
(3) demarcation edit matrix near error balancing point
(4) demarcation line at error balancing point

Method (1) is very close to that used in the reclassification neural network reported at NATDAC ‘96. Other variations of the demarcation line method were reported at the NATDAC ‘96 conference including the moving demarcation line. The edit matrix above is computed as a function of the ratio of N1:N2, and, based on the results above is expected to perform better than the moving demarcation model.

Uncertain Class 2

When we know the shape of the distributions but do not know a parameter such as the lateral location of class 2, constructing unbiased estimates is more difficult. Two parameters need to be estimated (i) the lateral displacement parameter of class 2 and (ii) the resultant population of class 1 (and that derived for class 2). Of the techniques described above, only the weighted least square technique is suitable for an estimate of the lateral location.

Here we give results for an analysis based on Model A, when the actual distribution is Model B. The demarcation line is chosen at 249.6-cm and the sample size is 100:60 as before for each of 1000 trials averaged. The displacement for class 2 in Model B is 4 grid units to the right of its location in Model A.

The least square estimate of population error results, based on finding the lowest least squares residual, has the lowest bias, if any. The residuals and biases are given in Table 5.

TABLE 5. Least Square Results

offset	*residual*	*bias*
6	46+-22	0.16+-1.1
5	39 +-19	0.2 +-1.1
4	40+-26	-0.04+-0.7
3	50+-34	-0.7 +-1.1
2	78+-61	-2.4+-1.2

Here the number following the +- is the standard deviation for the 160 events. Thus the residual for an offset of 5 is 39 with a standard deviation of 19. The value 39 is accurate to +- 19/31.6 ~ 0.6 where 31.6 is the square root of 1000. Thus it is not clear with these data whether an offset of 4 or 5 is given by the least square method; more trials would clarify this situation. At an offset of 5 this method gives a bias of 0.2 in the error, i.e., class 1 population is 100.2 +- 1.1 for the average 160 event trial.

By comparison the bias introduced by all techniques is shown in Table 6.

TABLE 6. Comparison of bias

method	*bias*
least square	0.2 +-1.1 or less
raw demarc	2.6+-1.6
bin. ed.	-5.3+-1.1
demarc. ed.	-10.8+-1.7

The 2.6 value for the raw demarcation line and the other values reported are not representative for all locations of Model B, just the one chosen. The point is that the location uncertainty induces an error that introduces a bias into the estimators.

The conclusion

The conclusion of this second phase of the study is that when there is known parametric variation in distributions of vehicles, the first step of a bias-free estimate of the class populations is the estimate of those parameters.

If data are to be taken for a long period of time at a given site, it may be possible to make estimations of parametric distributions.

A barrier to using this method now is that binned axle separation data is not collected as a matter of course, and the additional storage such data would take requires a non-zero cost. Given such data of sufficient quantity and quality however, much could be said about how distributions vary with site and time. The extent of ground truth studies required to verify the models in the overlap region could then be determined.

The rational for acquiring such data is *truth in data*: Obtaining that which is measured rather than the processed subjective non-standard measurements. Standard methods could then be developed for reducing the binned archived counts to estimates of vehicle class.

Even without established reduction models, such data could also have considerable archival value for developing such models and for the future when more is known about distributions. Such data could also serve as a check on the operation of the data collection process as a whole and would serve to resolve any disputes over published conclusions involving vehicle statistics.

Footnote

This problem can also be viewed as a degenerate case of a Hidden Markov Model with a bin-label alphabet. [See L. R. Rabner, Proc. IEEE 77, 257-286, 1989, for a tutorial on the Hidden Markov Model]. The solution for that approach reduces to a maximum likelihood estimate of the offset, perhaps different from this least square approach.

8/96

Vehicle Reject via PNN

The probabilistic neural network (PNN) based on hash tables described for the NATDAC conference is the technical basis of monitoring vehicles rejected by that net. Next month we will develop a test and determine its usefulness by applying it to vendor data.
Our NATDAC measure of vehicle rejection is expected to be relatively independent of vehicle distributions and their changes with time. With sufficient input data, the vehicle distributions can be made insensitive to location. We may apply this rejection measure to hourly vendor data and to data accumulated in the Idaho Classlog studies of 1992 and 1993.

We will read in the vendor data file along with the corresponding ground truth file. If overall evaluation of the test goes well, our preliminary plans are as follows: We will obtain an hour’s worth of data at a time. We will classify to see if it is rejected or not. We will record the results in a file containing: the beginning hour time, the end time, and number of events in the hour, and the number of rejected events. This we will compare to the ground-truth rejection rate for the two test files.

10/96

Neural Network for arbitrary dimension

The simple description of neural networks we gave in the spring summary illustrated their potential in describing an arbitrary two-dimensional function. How the weights are obtained in the network, however, was not specified.

This situation is analogous to a theorem based on work by Hecht-Nielsen[1987], Sprecher[1965], and Kolomogorov followed by Gallant and White[1988] and Cybneko[1988]. The theorem stated in neural network language is: *any N-Dimensional function of finite support can be constructed with a three-layered neural network*. Finite support indicates a bounded interval in which the function is defined.

This theorem does not provide an algorithm for training the neural network. References and summary are to be found in a comprehensive book entitled **Neural Networks** by Simon Haykin [1994].

Training a Single-Layer Perceptron Classifier

Here we will discuss the properties of single-layered perceptrons. We train a perceptron with data that falls into two classes. The data must meet the following requirement: it must be divisible by a planar surface so that one class of data is on one side of the plane and the other class is on the other side. We have proven that these perceptrons converge for all fixed choices of the learning parameter. This proof clears up ambiguities in an already known proof by Haykin 1994. Our proof works directly in the training example space known as the exemplar space rather than in a scaled or distorted version of it.

Our explanation of these concepts has a minimum of algebra. We train the dividing plane of the perceptron to reorient itself until it divides the two data classes. This dividing plane is characterized by a set of weights that may be thought of as a vector perpendicular to the plane. The nullified region is the portion of weight space wherein the division of the two data classes is complete and the weights do not change for any training data.

The perceptron produces an output of plus or minus unity that corresponds to the two classifications of the training data. For example, we may think of data as being either in the class of true giving output of +1 or in the class of false having output of -1. Each of the N sets of unique training exemplars has P continuous variables (a point in P-dimensional space) and the corresponding classification class. Each point is represented by P real numbers enclosed in parentheses and separated by commas. The set of N exemplars or points presented to the network are called an *epoch*. Multiple epochs are repetition of such a presentation sequence.

Higher-dimensional points

Because of our diverse audience we will describe one example of a possible five-dimensional space. We use a speck of dust to represent a point. We assign the fourth dimension to be time. The fifth dimension is the color of the speck.To make each dimension measurable in centimeters we multiple the time by a speed given as 1-cm/s so that one second represents a single centimeter. We convert the color to centimeters by means of an imagined rainbow that has a fixed distance between two colors. Image this rainbow straightened out, as if painted on a thin beam of wood with the colors starting at the upper edge of the beam and extending to the lower.

One easily imagined three-dimensional space has a horizontal length, a vertical color, and a depth of time. If we want to describe the height of the speck above the floor in our imagined space, we who live in the real three-dimensional world are out of luck because we are using height for the color. So if we agree to the game of calling height color and we know a speck is 5-cm above the floor, we can determine the color of the speck by measuring on the beam painted like a rainbow; however, we cannot say the real height of the speck above the floor. We use similar games to deal with the fourth and fifth dimensions.

To describe the distance in centimeters between two specks in five-dimensional space we first determine where each speck is in that space. The location of the speck is described by five numbers, each would be measured in centimeters from the five walls in five-dimensional space. Two separated specks are described by two sets of five numbers. If four corresponding pairs of numbers describing the two specks are the same but the fifth pair of numbers differs by 10-cm, the two specks are 10-cm apart irrespective of which dimension contains the separation.

Another similar rule of the game is that if two or three corresponding pairs of numbers differ but the remaining two pairs are the same, we disregard the matched pairs. Five numbers with two or three struck out represent points in three or two dimensional space, respectively. The unmatched numbers are then the distances in centimeters from the three or two remaining walls, respectively. In the usual way we position the two specks in three-dimensional space and determine the distance between them. This procedure is familiar if the three numbers represent the horizontal width, vertical length and the depth of ordinary space and if specks have equal time and color.

Rather than measure the distances, we know from geometry that we can use the two-dimensional theorem of Pythagoras to give us the same information. In two dimensions we apply the theorem once. In three dimensions we apply it twice. The rules of the game are that if four numbers differ we apply it three times, and if all five numbers differ we apply it four times.

We can summarize the rules we have discussed so far in the single rule for determining distance between two specks: take the square root of the total sum of the squares of the differences in each of the five dimensions.

Perceptron training

The Rosenblatt fixed threshold theorem for convergence of the training algorithm for single-layer perceptrons [1961] applies when the data are linearly separable. It follows, in our example, that none of the data in the two classes overlap. A special requirement is that any two training exemplars that have the same values of training variables (i.e. two identical sets of five values in five dimensional space) must have the same classification. If this were not the case, in anthropomorphic terms, the network could argue with itself unceasingly — a condition we call *caterwauling*. For example, on the first epoch we could train the weights to correspond to the class of true and then the next instance to the class of false. On each subsequent epoch the weights could repeat the process, never ceasing to change and never converging. Because of the separability requirement we have ruled out this type of behavior.

Anthropomorphic terms are appropriate to neural networks because the operation of the human brain has many aspects in common with a neural network. Perceptrons do not compromise. If any point is not classified correctly, the perceptron changes its weights, thereby moving the decision plane until the point is correctly classified. Exactly how the perceptron learning machine does this is the subject of the rest of this description.

During the presentation of a single training point, the perceptron strengthens or weakens its weight parameters. For now, the weight parameters can be thought of as a point or vector (from the origin to that point) in the exemplar space. The weights represent the state of the training. This weight vector is perpendicular to the plane used to divide the data. The change in the weight vector depends only on three conditions: (1)correctness of classification (2) the size of the learning parameter and (3) the position of the misclassified point in *P*-dimensional space. With correct classification the need for the parameters to change is nullified, and the point remains at rest. The rule described more fully below is that in the case of any misclassified point, the perceptron modifies the weight components (e.g. five components in a five dimensional space) by adding to them the corresponding products of each of the point's dimensional components with the positive learning parameter.

According to Anderson, Nilsson introduced a transformation permitting variable thresholds. This transformation embeds a point from *P*-dimensional space into (*P+1)*-dimensional space where the additional dimension always has the value of negative unity for the training variable. For example, P=2 and the ground plane represents the space of the training variables prior to applying the Nilsson transformation. Then the transformed points are placed one unit underground rather than on the ground plane. Afterwards, when convergence is achieved, the trained plane passes through the origin and separates the points into their two classes. The intersection of the trained plane with the underground plane is a line offset from the origin. This simple artifice complicates the problem by adding a dimension, but it simplifies the problem conceptually because the training plane is no longer free to move away from the origin. The offset of the separation line in the underground plane is now governed by the tilt of the separation plane passing through the origin and which is not offset from the origin. Rosenblatt[1961] apparently realized that the concept of offset was not an essential part of his argument and simply considered the offset to be fixed at an ideal location. By moving the description to one dimension higher, Nilsson removed the need to consider every possible offset.

It is clear that the weight components in this (P+1)-dimensional space constitute a vector (another name for a straight line or stick with a head and a tail), $\boldsymbol{w}$. We can consider a point in any-dimensional space to be a vector by using the point for the head of the vector and the origin for its tail. Two vectors are said to be "allied" if the projection of the first vector onto the second is in the direction of tail to head, or origin to point. When a point, which is considered a vector, in the *P+1* dimensional space is allied with $\boldsymbol{w}$ it is, by definition, correctly classified; if it is not allied with $\boldsymbol{w}$**,** it is incorrectly classified. When the training point is not allied with $\boldsymbol{w}$, $\boldsymbol{w}$ is changed component-by-component by an amount that is the product of the learning parameter and the training point regarded as vector $\boldsymbol{x}$. Thus, the following is the single class learning rule: *if* $\boldsymbol{x}$ *is not allied with* $\boldsymbol{w}$*, move* $\boldsymbol{w}$ *in the direction of* $\boldsymbol{x}$ *an incremental amount that is the product of* $\boldsymbol{x}$ *with the positive learning parameter*. Add this increment by vector addition: by laying the tail of the attenuated (or magnified) $\boldsymbol{x}$ vector at the head of the $\boldsymbol{w}$ vector. The location of the incremented $\boldsymbol{w}$ vector is then at the head of the $\boldsymbol{x}$ vector.

For a single exemplar, the correct convergence of the network should be clear. If $\boldsymbol{w}$ has allied with $\boldsymbol{x}$, no change occurs, and the network has converged. If $\boldsymbol{w}$ has not allied with $\boldsymbol{x}$, then on each subsequent epoch the multiple of $\boldsymbol{x}$ is added to $\boldsymbol{w}$ until it is allied with $\boldsymbol{x}$. Then, the network reaches the nullified spot.

We now give a single-class example of caterwauling using two exemplars. We achieve the lack of linear separability by placing the first point $\boldsymbol{x}$ in (*P+1)*-dimensional space and the second point at the location $-\boldsymbol{x}$. A negative vector is obtained by fixing the tail at the origin and moving the head directly toward the origin and beyond an amount that makes the final length the same as the original length, but the vector is pointing in the opposite direction. Then almost everywhere we happen to choose $\boldsymbol{w}$, one of the two points, $\boldsymbol{x}$ or $-\boldsymbol{x}$, is not allied with $\boldsymbol{w}$. That misclassified point causes $\boldsymbol{w}$ to take one step per epoch toward the misclassified point. When $\boldsymbol{w}$ is finally allied with the misclassified point, the subsequent step exactly reverses the previous step, so that $\boldsymbol{w}$ oscillates back and forth non-deterministically between those last two values.

The general convergence of the single-layer perceptron assumes that the linear separability condition occurs and must be proven for only the single-class case. We transform the two-class case to a single class case using another idea suggested by Nilsson 1965. We can reflect any point in Class 2 at location $\boldsymbol{x}$ through the origin to the point $-\mathbf{x}$ and assign it to Class 1. The Class 2 points are then above ground level by one unit cater-cornered from their old locations, whereas the Class 1 points remain one unit underground. That transformation does not affect conditions at all on the allowable separation plane position, which must contain the origin.

Part of the proof involves establishing a small neighborhood for a particular weight vector, labeled $\boldsymbol{w_0}$, in which any member vector w of that neighborhood has the same property defined for w_0 (i.e.all the training exemplars are in its direction, and so no corrective steps are required). We know there must be at least one w_0 that is normal to the plane that contains the origin and gives the assumed linear separability. We call any w in the previously mentioned neighborhood, *a "*classifier vector" and the largest such region made up of all classifier vectors, which includes $\mathbf{w_0}$, *the nullified region*.

Our knowledge about the existence of the small neighborhood follows from the fact that the training points are not infinitely dense. Any candidate separable plane that goes through the origin could contain some of the training points, but all of the other training points are a finite distance away. This finite distance might be small, but it is nonzero. Thus, we can rotate the candidate plane ever so slightly in any direction without encountering any of the points. This means that when we rotate the plane from w_0 to any w that is within the small rotation of w_0, the plane crosses none of the training points. Hence, no allies change sides during these movements, and this entire region is contained in the nullified region of w_0. Furthermore, any positive extension or attenuation of w has the same direction as w and these changes solely in length will not cause allies to change sides. Thus, a cone-like region that includes attenuated weight points arbitrarily close to the origin and extended weight points infinitely far away from the origin constitute the nullified region.

The finite step size aspects are considered next. Each exemplar produces only one step size for or increment to w, which is used to modify w when the exemplar is incorrectly classified. This condition occurs irrespective of the fixed size of the learning parameter. Also the nullified region contains a sphere of finite albeit small dimension R centered on w_0. If the length of the maximum step size is Z, and the length of w_0 is $|w_0|$, it follows that the nullified region also contains a sphere of radius $Z+1$. Any point on this larger sphere is obtained by moving in the direction from the origin to the sphere of radius R and extending beyond a fixed multiple of the distance from the origin to the point on the sphere. All these points beyond the sphere are in the nullified region. If we choose any positive multiple, the magnification ratio, and choose all the points on the small sphere, we obtain a new sphere whose radius is also magnified by that ratio. By choosing the magnification large enough, we conclude that the nullified region must also contains a sphere of radius $Z+1$. The sphere is centered on a multiple of the vector w_0, we call w_1. The magnification of w_0 that will give w_1, is $(Z+1)/R$, thus, giving $w_1=w_0(Z+1)/R$. Greater magnifications also suffice. The point w_1 has been constructed such that any single step from outside the nullified region can not reach w_1 within the nullified region. Although aspired to, point w_1 can never be reached.

The nullified region is a pencil of rays going through the origin, but excluding the origin. We now demonstrate that the pencil boundary must be made up of N planes each of which contains the origin. There is one plane associated with each exemplar. Each exemplar is a point, x_i, in $P+1$ dimensional space. We construct a plane containing the origin whose normal is in the direction of x_i for each of the N exemplars. We call the side of each plane containing x_i *positiv*e. Then the points that are positive with respect to all planes are said to be in the intersection (overlap) of the positive regions. This region with the origin removed, is the nullified region.

The intersection of one or more positive sides of planes is called a "convex" region. In addition, all planes have the origin in common, and we call this region a (planar vertex-free) convex cone, which is a $P+1$ dimensional pyramid with a bottomless base and no vertex. If a weight point is not part of the convex cone and an exemplar is presented that is not allied with w, w takes a step in the direction of x_i, which is the normal direction to the plane associated with x_i, and toward the inside of the cone. The step is also toward the plane. It does not exceed the boundary of the plane if the step size is small enough. In this case the step moves w toward w_1 each on opposite sides of the plane. If the step happens to be large enough, it could step inside the convex cone or even to the other side of the convex region. Even if it does this, it is still moving closer to the point w_1 because w_1 was chosen to be more than a step away from the planar convex cone boundary of the nullified region.

The point $\boldsymbol{w}$ is a finite distance away from $\boldsymbol{w_1}$. Each exemplar not in the direction of $\boldsymbol{w}$, moves $\boldsymbol{w}$ a step closer toward $\boldsymbol{w_1}$. Each such step moves a finite distance bounded below, and, therefore, convergence will require at most a finite number of steps. This convergence property occurs irrespective of the size of the learning parameter, although the number of required steps will depend on its value. Too large of a value will cause large jumps across the nullified region, from outside to outside. What we have shown includes cases starting near the cone vertex that result in the moving upward of $\boldsymbol{w}$ from the region of the cone vertex toward $\mathbf{w_1}$, jumping across the cone each step, to eventually converge when jumping within the nullified region.

Our presentation is intended to clear up some confusion about requirements on the allowable magnitude for the fixed learning parameter: it can be positive and arbitrarily large. Lippmann 1987 calls the learning parameter the “gain fraction”. However, contrary to our result, he states without proof that the parameter eta “*is a positive gain fraction less than 1”*. This result which has been quoted by Haykin 1994 should not have the bound of 1 on eta for convergence. We have shown eta can be unbounded and convergence will still occur.

References

Anderson, J. A., 1995, An Introduction to Neural Networks.

Haykin, S., 1994, Neural Networks, A Comprehensive Foundation, Saddle River, NJ, Prentice Hall.

Lippmann, R. P., April 1987, *An Introduction to Computing with Neural Nets*, IEEE ASSP Magazine.

Nilsson, N.J., 1965, *Learning machines: Foundations of Trainable Pattern-Classifying*, New York: McGraw Hill.

Rosenblatt, F., 1961, Principles of Neurodynamics, Washington D.C., Spartan Books